(19) JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **05307440 A**

(43) Date of publication of application: **19.11.93**

(51) Int. Cl **G06F 3/06**
G11B 20/12

(21) Application number: **05009002**

(22) Date of filing: **22.01.93**

(30) Priority: **06.03.92 JP 04 49833**

(71) Applicant: **MITSUBISHI ELECTRIC CORP**

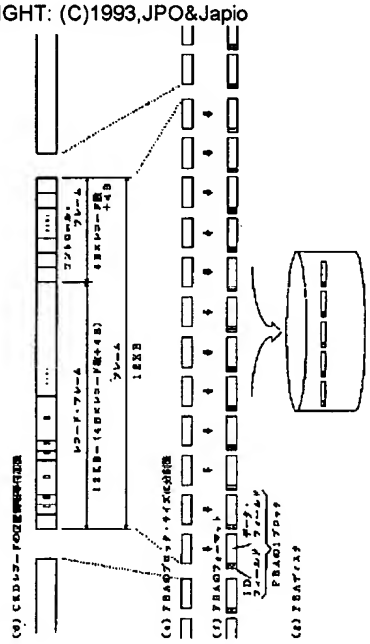
(72) Inventor: **NAKAMURA YOICHI**

(54) **DATA STORAGE FORMAT CONVERSION SYSTEM AND ITS CONVERSION METHOD, ACCESS CONTROLLER AND DATA ACCESS METHOD**

(57) Abstract:

PURPOSE: To provide a record storage format conversion system by which the number of memory access times for detecting a target CKD record is less, a memory load is reduced and memory capacity for holding track data is little at the time of format-converting the CKD record into an FBA record.

CONSTITUTION: All the gaps such as an inter-record gap and an inter-field gap are eliminated from one track in the CKD record. The track data is divided into frames being management units, which are equivalent to integer times as large as the fixed block length size of the FBA record. Control information showing the positions of all the CKD records included in the management unit are written into the last end of the management unit in order from the rear as control frames, and they are divided into the fixed block length size of the FBA record. Then, information showing a relative position from the head of the track at the CKD record format are held in the respective CKD records.



* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] In the data storage format conversion method which changes and memorizes the record memorized by the variable length format to a fixed-length format A gap deletion means to delete an inter-record gap and the gap between the fields from the variable-length record for one truck recorded by the variable length format, and to generate truck data, The 1st positional information which shows the location of the variable-length record which arranges the truck data which deleted the gap with the above-mentioned gap deletion means per management of predetermined size, and is contained in the management unit, A record layout means to memorize the 2nd positional information which shows the relative position from the head of the truck in the variable-length record format of the variable-length record contained in the management unit per [above-mentioned] management, The data storage format conversion method characterized by having a fixed block storage means to divide and memorize the above-mentioned management unit in the fixed length block length size of fixed-length record.
[Claim 2] The above-mentioned record layout means is a data storage format conversion method according to claim 1 characterized by arranging the 1st and the 2nd positional information corresponding to the arranged variable-length record sequentially from another side of the above-mentioned management unit while arranging a variable-length record sequentially from one side of the above-mentioned management unit.
[Claim 3] The above-mentioned record layout means is a data storage format conversion method according to claim 1 characterized by appointing beforehand the field which memorizes the 1st and the 2nd positional information, and the field which memorizes a variable-length record in a management unit.
[Claim 4] It is the data storage format conversion method according to claim 1 which the 1st positional information of the above is a memory address within management single [of the variable-length record contained per management], and is characterized by the 2nd positional information of the above being the sector value or segment value from a head of a truck.
[Claim 5] The above-mentioned variable-length record is a data storage format conversion method according to claim 1 characterized by having the information by segment which shows the relative position from the head of the truck in a variable-length record format in a variable-length record.
[Claim 6] The address information which shows the location of the variable-length record which deletes a gap from a variable length format, arranges the data per management of predetermined size, and is contained in the management unit, The relative-position information which shows the relative position from the head in the variable length format of the variable-length record contained in the management unit is made to hold. An R/W means to write data per [above-mentioned] management to the access-control (equipment a) above-mentioned storage section accessed using the following elements to the storage section which divided into the fixed-length block size of fixed-length record, and was made to memorize as

a fixed-length format, (b) Memory which stores the data written by the above-mentioned R/W means for every management unit, (c) An input means by which a record inputs the access instruction to the variable-length record outputted as what is memorized by the variable length format, (d) A location count means to estimate the location of the management unit considered that the variable-length record to access is memorized based on the access instruction inputted with the above-mentioned input means, (e) A record retrieval means to search the record which the access instruction which read into the above-mentioned memory the management unit in the location estimated by the above-mentioned location count means from the above-mentioned storage section with the above-mentioned R/W means, and was inputted by the above-mentioned input means tends to access.
[Claim 7] The above-mentioned access-control equipment is access-control equipment according to claim 6 characterized by judging an operating capacity of a truck in case the record is memorized by the variable-length record method further using relative-position information.
[Claim 8] While the above-mentioned input means inputs the relative-position information on the record which it is going to access, the above-mentioned record retrieval means Compare the relative-position information currently held at the searched management unit in the relative-position information on the record which is going to carry out [above-mentioned] access, and until a management unit with the relative-position information on the record which it is going to access is found A management unit retrieval means to search a management unit by reading the following management unit into memory from the storage section with an R/W means, When a management unit with the relative-position information on the record which it is going to access with the above-mentioned management unit retrieval means is found Access-control equipment according to claim 6 characterized by having a record specification means to specify the variable-length record which it is going to access using the address information contained in the management unit.
[Claim 9] An inter-record gap and the gap between the fields are deleted from the variable length format which has the field and the inter-record gap which have the data storage format conversion approach (a) data which have the following processes, and a gap between the fields. While arranging in order on a frame with the size which was able to define beforehand each variable-length record connected by the gap deletion process and the (b) above-mentioned gap deletion process which connect the field of each variable-length record The record layout process which each variable-length record is made to correspond and memorizes the access information for accessing each variable-length record, the fixed length block storage process which divides the (c) above-mentioned frame into two or more fixed-length block sizes, and is memorized as a fixed-length format.
[Claim 10] The above-mentioned record layout process is the data storage format conversion approach according to claim 9 characterized by memorizing the positional information of the frame by which each variable-length record has been arranged, and positional information in case each variable-length record is memorized based on the variable length format as access information.
[Claim 11] It has the following processes. By the data storage format conversion approach of the claim 9 above-mentioned publication The access command input process of inputting the retrieval information on the command for accessing the record of the data-access approach (a) variable length format which accesses the data of the memorized fixed-length format, and the data to access, (b) The positional information presumption process of presuming the positional information of the frame the record accessed from the retrieval information inputted according to the above-mentioned access command input process is remembered to be, (c) Record search procedure which reproduces the frame after a frame with the positional information presumed according to the above-mentioned positional information presumption

process from two or more fixed length blocks memorized in the fixed-length format in order, and searches the record to access.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Industrial Application] This invention relates to a data storage format conversion method for the fixed-length method used by the commercial small disk unit to realize the variable-length record method used with the magnetic disk drive of a general-purpose computer etc.
[0002]
[Description of the Prior Art] Conventionally, as external storage of a general purpose computer system, the hard disk unit of a variable-length record (CKD;Count Key Data) method is used widely. A variable-length record method is a record format the die length of the physical record recorded on one truck on a magnetic disk and whose number are adjustable. Generally, one truck of the magnetic disk of a variable-length record method begins from an index mark (not shown), as shown in drawing 33 , and the home address (HA) and the continuing record zero (R0) exist first. In addition to address information, such as a cylinder address of that truck, and the head address, if this HA and R0 are the location of the magnetic blemish on that truck, the information on whether it is a defective track, and a defective track, since it includes the control information of the address a phase hand's truck by which shift allotment is carried out etc., it must exist. By the record by which the record after the record 1 (R1) which follows this HA and R0 records the usual data, as long as one-piece record [one piece] die length and the number are the capacity which that total capacity can record on a truck, it can be used, formatting for the software (S/W) on a host computer freely. Each record consists of the three fields, the count field (COUNT), key field (KEY), and a data field (DATA), and is called a CKD method from the initial of these fields. The die length of the count field is immobilization, and since the die length (KL) of the key field of the record, the die length (DL) of a data field, etc. are included other than address information, such as a cylinder address (CC), the head address (HH), and a record number (R), the die length of the continuing key field and a data field is freely changeable. Although key field and a data field are the parts which record the actual data of the record, when the key field which should record the key of the record are unnecessary, the record with which key field do not exist can also be made by specifying KL=0 in the count field. HA, R0, R1, and R2 -- between ... etc. the inter-record gap (G1, G2, G3) of suitable size prepares -- having -- **** -- especially -- R1 and R2 -- about the gap in front of records, such as ..., (G3) Since [which is called the address mark (AM)] the special mark is recorded magnetically, even if it begins to read from which part in a truck, the reading start of a record can be found by detecting this address mark.
[0003] Moreover, in order to distinguish each field also inside each record, the gap between the fields (G2) is prepared, respectively. On the other hand, as external storage of comparatively small-scale computing systems, such as a minicomputer, an office computer, a personal computer, and a workstation, the hard

disk unit of a fixed-length-record (FBA;Fixed Block Architecture) method is used. By the disk of a fixed-length-record method, one record (by the disk of a fixed-length-record method, called a sector in many cases) usually consists of the ID field which is recording address information, and the data field which records data, and any die length of the field is immobilization. Therefore, the record count around one truck (the number of sectors) is also immobilization.

[0004] Conventionally, the above proper use has been performed by the target computing system [magnetic disk / the magnetic disk of a variable-length record method, and / of a fixed-length-record method]. Therefore, since the target system was small, compared with the disk of a variable-length record method, generally the disk of a fixed-length-record method was small, its memory capacity was also small, and its engine performance, such as a data transfer rate and a seeking rate, was also low. However, conversely, that a demand of as opposed to a miniaturization also in the disk of a variable-length record method is becoming remarkably high, and since the direction of the disk of a fixed-length-record method has large-capacity-izing and remarkable improvement in the speed, the difference among both in a specification side is becoming small in recent years. On the other hand, as for the disk of a variable-length record method, since the direction of a price is exclusively for a general-purpose computer, even an office computer is inferior to volume efficiency in it compared with the disk of the fixed-length-record method used widely from a personal computer. Therefore, it becomes surely comparatively high-priced and the gap of both cost performance is spreading. For this reason, the demand referred to as wanting to use the disk of a cheap and highly efficient fixed-length-record method also with a general-purpose computer has become strong.

[0005] Moreover, in the hard disk unit of a CKD method, the servo surface servo system which prepares separately the data surface which records data, and the servo surface which records the servo information for positioning a head has been used. The positioning accuracy of a head will also be prescribed by the precision of the mechanical location of a data surface and a servo surface by this method. Then, in order to attain densification of a truck further, the data surface servo system which also writes servo information in the ID field in a data surface etc. has come to be used. If this data surface servo system is used, extremely, by the variable-length record method allowed 1 truck 1 record, there will be no security that servo information is always acquired at fixed spacing, and it will become disadvantageous about the positioning accuracy of a head compared with the fixed-length-record method with which servo information is surely acquired at fixed spacing. Furthermore, since there is a limitation in the one-set disk [one set] improvement in the engine performance, or improvement in dependability, Although the technique of the disk array which distributes and records data on two or more disks, or prepares the redundancy disk which records the parity between the dispersed data etc., and aims at improvement in the engine performance and dependability attracts attention It is also difficult to deal with such a technique, while the die length and the number of a record have been the variable-length record method which is adjustable. however, software property huge in the world of a general-purpose computer -- **** -- him -- for a ****** reason, it is remarkably difficult to change a record format of a disk into a fixed length from variable length.

[0006] Here, the access method to the variable-length record recorded by the conventional variable-length record method using drawing 34 and drawing 35 is described. Drawing 34 is drawing showing typically the magnetic disk with which the servo surface servo system is used, and servo surface S and data surface D are prepared in this magnetic disk drive. The sector (sector of a different concept from the sector of a FBA method) is logically prepared in servo surface S. Drawing 35 is drawing for explaining this sector. In the storage capacity of one truck, supposing it is 48KB, this one truck will be divided into two or more sectors here. Here, size of 1 sector is set to 244B. This 1 sector consists of seven segments called segments S1-S7

further. Since the size of 1 sector is 244B, one segment is set to 32B. The value of the sector on which data were recorded when data were recorded on data surface D using such a magnetic disk drive is memorized, and in searching the data, it searches using the sector value when being written to a data surface. It becomes possible to search the data with carrying out like this at a high speed. Drawing 36 is drawing showing an example of the access sequence from software which operates with the host computer in the case of searching data using a sector value. The software which operates with a host computer as shown in drawing 36 specifies the sector on which the data which it is going to access are recorded with a set sector command. A magnetic disk drive is separated from a host computer and disk storage control by this set sector command, and the specified sector is searched autonomously. If the specified sector is discovered, a magnetic disk drive will advance a re-connection request to a host computer through disk storage control, and re-connection will be made. Next, ID of the data which should be searched with a search ID command is sent. this ID -- for example, a cylinder number and a track number -- or it is the record number which should be searched. If this search ID command is received, a magnetic disk drive will compare ID of the record which exists in the beginning from the sector specified by that ID and the set sector command, and will report that result to a host computer. If both ID is not in agreement, activation of a command is jumped and returned to a set sector command by the following transfer in channel command, and a search ID command is again published to the following record. If both ID is in agreement, a host computer will execute an instruction called a lead or a light to the searched data by flying the following transfer in channel command and publishing the last lead data command or light data command.

[0007] Thus, it is remarkably difficult to already have created the software used with a general-purpose computer based on the variable-length record method, and to change the already built software. Then, the actual disk is considered from the software on a host computer by emulating a variable-length record method with disk storage control in the technique which can use the disk of a variable-length record method as usual, using the disk of a fixed-length-record method.

[0008] CKD-FBA conversion is stated to the detail in "the storage control approach and equipment" (JP,1-30691,A). Moreover, still more generally CKD-FBA conversion is indicated to the publication "IBM4321 / 4331 processor compatibility facility" (GA33-1528, the 3rd edition, September, 1982) of IBM. It is as follows when the contents of these reference are summarized. With the above-mentioned publication, in case it changes into FBA from CKD, all gaps are taken and compressed. Moreover, it is described by the above-mentioned publication that the set sector command shown in drawing 36 is not supported. When a set sector command is taken out with the above-mentioned publication, it is indicated that it becomes no operation. When the set sector command is not supported, record retrieval will be performed from a search ID command. Although a record can be searched only with a search ID command, in order to search the record of the same ID as ID specified by the search ID command, looking at all records in order from the head of the truck, retrieval time will start. The set sector command is very disadvantageous in respect of the access time compared with the case where the sector or later which specifies the familiar sector in which the record exists and was specified by the set sector command is searched with a search ID command.

[0009] on the other hand -- the above-mentioned official report -- (-- the number in a parenthesis takes correspondence), the gap between (1) FIRUDO (G2) and ECC, padding, a physical parameter, etc. to drawing 37 .

(2) An inter-record gap (131,133,137,163) leaves.

Only the part with the same parts, such as a gap between the fields taken by (3) and (1), lengthens the inter-record gap of (2). Thereby, the relative position from the head of a truck to the count field of each

record is maintained with origin. (Although the relative position of key field or a data field will shift, if even the relative position of the count field is known, positioning of a record will not be hindered by it) .
(4) Divide this truck data into the fixed block (sector) of FBA, and add the positional information (156) of the count field of the first record contained in each fixed block to the head of each fixed block in that case. Thereby, direct and random addressing to the CKD record memorized by the disk of a FBA method becomes possible.
(5) When the count field is not included in the block, record the flag bit which shows it.
If it furthermore arranges, in order to maintain the relative position of count FIRUDO of (1) each record, it will leave only an inter-record gap, and other excessive gaps etc. will be removed.
(2) Attach to the head of each fixed block of FBA the header which shows the location of the first count field currently recorded there.
Two ** serve as requirements and direct and random addressing to the CKD record recorded on the disk of a FBA method is made possible by carrying out like this.
[0010]
[Problem(s) to be Solved by the Invention] However, there is the following trouble in the advanced technology mentioned above. In case it changes into FBA from CKD according to the technique by the publication mentioned above, although all gaps are taken and compressed, since the set sector command is not supported, access of a record takes long duration. Moreover, especially, although an inter-record gap deletes in the technique shown in the official report mentioned above, after leaving, in order that only the part of the eliminated inter-record gap may lengthen the gap between the fields, when [that a record size is small] there is much record count, futility of the gap between the fields increases on memory.
[0011] Moreover, since it has only the information about the location of a top CKD record among the CKD records contained in each fixed block (record of a fixed-length-record method) of FBA In order to find CKD records other than a head Read the count field of a top CKD record first, and the top record length is got to know from KL in it, and DL. After calculating the location of the count field of the following record, KL of the count field of the following record and DL must be read, and the procedure of looking for the count field of the following record must be repeated. Therefore, when the centralized control of the truck image is carried out on the disk cache etc., the count of access to the memory increases remarkably, and the processing time not only starts, but it reduces the system-wide engine performance. Furthermore, since it has the positional information of a CKD record in the small unit of every FBAblock, at the time of data transfer with a channel, interruption occurs frequently and causes aggravation of data transmission efficiency temporarily [data transfer] for flying and exceeding this positional information.
[0012] This invention aims at obtaining the record storage format conversion method with which there is little memory space for holding truck data, and it ends while there are few counts of memory access for discovering the CKD record made into the purpose, and they end and memory unloading is made, in case it was made in order to cancel the trouble in the conventional example mentioned above, and format conversion of the CKD record is carried out to a FBA record.
[0013]
[Means for Solving the Problem] The data storage format conversion method concerning this invention All gaps, such as an inter-record gap and a gap between the fields, are deleted from one truck of the variable-length record which is the storage format used for the external storage of a general purpose computer system. The truck data is divided into the management unit equivalent to the integral multiple of the fixed-length block size of fixed-length record. After writing in the relative-position information which shows the relative position from the head of the truck in the variable-length record format of address information and

each variable-length record which shows the location of all the variable-length records contained in the management unit per [above-mentioned] management, It is characterized by what is divided and memorized to the fixed-length block size of fixed-length record.

[0014] Moreover, the access-control equipment concerning this invention is access-control equipment which accesses to the storage section which memorizes the data created using the data-storage format-conversion method mentioned above, it interprets the access instruction of the variable-length record taken out from the software created based on the conventional variable-length record method, accesses the record memorized by said fixed-length-record method, and has the following elements.

(a) An R/W means to write data per [above-mentioned] management to the above-mentioned storage section, (b) Memory which stores the data written by the above-mentioned R/W means for every management unit, (c) An input means by which a record inputs the access instruction to the variable-length record outputted as what is memorized by the variable length format, (d) A location count means to estimate the location of the management unit considered that the variable-length record to access is memorized from the access instruction inputted with the above-mentioned input means, (e) A record retrieval means to search the record which the access instruction which read into the above-mentioned memory the management unit in the location estimated by the above-mentioned location count means from the above-mentioned storage section with the above-mentioned R/W means, and was inputted by the above-mentioned input means tends to access.

[0015] Moreover, the data storage format conversion approach concerning this invention deletes a gap from the variable-length record created by the variable-length record method, makes the access information of the location of the data arranged while having arranged on the frame which is the management unit which was able to define this beforehand etc. memorize, divides this frame into a fixed-length block size, memorizes it as a fixed-length format, and has the following processes.

(a) The gap deletion process which deletes an inter-record gap and the gap between the fields from the variable length format which has the field and the inter-record gap which have data, and a gap between the fields, and connects the field of each variable-length record, (b) While arranging in order on a frame with the size which was able to define beforehand each variable-length record connected by the above-mentioned gap deletion process The record layout process which each variable-length record is made to correspond and memorizes the access information for accessing each variable-length record, the fixed length block storage process which divides the (c) above-mentioned frame into two or more fixed-length block sizes, and is memorized as a fixed-length format.

[0016] Moreover, the data-access approach concerning this invention is an approach for accessing with the access instruction of the variable-length record method which exists from the former to the data divided into the fixed-length block size created by the data storage format conversion approach mentioned above, and has the following processes.

(a) The access command input process of inputting the retrieval information on the command for accessing the record of a variable length format, and the data to access, (b) The positional information presumption process of presuming the positional information of the frame the record accessed from the retrieval information inputted according to the above-mentioned access command input process is remembered to be, (c) Record search procedure which reproduces the frame after a frame with the positional information presumed according to the above-mentioned positional information presumption process from two or more fixed length blocks memorized in the fixed-length format in order, and searches the record to access.

[0017]

[Function] In the data storage format conversion method in this invention, in order to delete a gap with a

gap deletion means, when holding data to a magnetic disk etc., there is little storage capacity and it ends. Moreover, since the 1st positional information a record layout means indicates the location of a variable-length record to be in a management unit, and the 2nd positional information which shows the relative position at the time of being recorded as a variable-length record are memorized, When the access instruction of variable-length record ****** comes from the conventional software, It becomes possible to search the target record using the 1st positional information which shows the location of the variable-length record in the management unit which searched the management unit using the 2nd positional information which shows the relative position of a variable-length record method, and then was searched. For this reason, the count of access can be managed with the minimum.

[0018] Moreover, in the access-control equipment in this invention, since a location count means calculates roughly the location of the management unit by the fixed-length-record method from the relative-position information on the record which accompanies the access instruction of the variable-length record method from the conventional software, a record retrieval means can access the data recorded by the fixed-length-record method based on this calculated positional information. Thus, when the access-control equipment concerning this invention is equipped with a location count means and a record retrieval means, it becomes possible to support a set sector command which was explained in the conventional example.

[0019] Moreover, after the data storage format conversion approach in this invention deletes a gap according to a gap deletion process, while it arranges each variable-length record in order in a frame according to a record layout process, the description is in the point of making a frame memorizing that access information both. Furthermore, since this frame is divided into a fixed-length block size and memorized, it can memorize a frame by the fixed-length-record method.

[0020] Moreover, in the data-access approach in this invention, the positional information in the case of memorizing based on a fixed-length-record method is presumed from the retrieval information as which the positional information presumption process was inputted by the access command, since a frame is searched based on this presumed positional information according to a record search procedure, it becomes possible to press down the count of access, i.e., the count of memory, which searches a frame to the minimum, and high-speed access is attained.

[0021]

[Example]

Example 1. drawing 1 is an outline block diagram for explaining this invention. The host computer 1 has the channel 2. The channel 2 has two or more channels 2a-2e, as shown in drawing. For example, disk storage control 5 is connected to one of channel 2bs of this. Disk storage control 5 has cache memory 6 inside. Disk storage control 5 performs a lead or light of data between the FBA disks 9 through cache memory 6. It block 9a-9e Has and all the data for which the FBA disk 9 has a fixed-length block size and that are memorized by the FBA disk 9 are written by reading and this block unit.

[0022] Next, the outline of CKD-FBA conversion is explained. In a host computer 1, the software for accessing the data recorded by the conventional CKD method operates. From this software, the CKD command 4 is outputted to disk storage control 5 through channel 2b. For example, when the CKD command 4 is a light command, the record 3 of a CKD format is outputted as variable-length records 3a, 3b, and 3c. Disk storage control 5 memorizes these variable-length records 3a, 3b, and 3c to cache memory 6. The truck image 7 of CKD is constituted by cache memory in that case. That is, the truck image 7 of CKD turns into an image which has arranged variable-length records 3a, 3b, and 3c sequentially from the head of a truck. Although the gap between the fields and an inter-record gap will be generated and added here if it is the former, all gaps are removed in this case. Disk storage control 5 changes into the record 8 of a FBA

format the truck image 7 of CKD made by cache memory 6. The record 8 of a FBA format is constituted by the fixed-length blocks 8a-8e. Disk storage control 5 outputs the FBA command 10, when writing the record 8 of this FBA format in the FBA disk 9. Blocks 8a-8e are memorized corresponding to the blocks 9a-9e in the FBA disk 9, respectively.

[0023] Drawing 2 is a block diagram for explaining in more detail the outline of the CKD-FBA conversion mentioned above. A host computer 1, disk storage control 5, cache memory 6, and the FBA disk 9 are the same as that of drawing 1 mentioned above, and omit the explanation here. Detailed actuation of this disk storage control 5 is further explained using drawing 3 , drawing 4 , and drawing 5 .

[0024] Drawing 3 , drawing 4 , and drawing 5 explain a record storage format conversion method. First, to the original CKD track format shown in drawing 3 (a), as shown in drawing 3 (b), all the gaps G1 and G2, such as an inter-record gap and a gap between the fields, and G3 are deleted.

[0025] Next, including the control information mentioned later, as 1 truck data after the gap deletion shown in drawing 3 (b) (for example, the CKD record which has 48KB) are shown in drawing 3 (c), it divides including the management unit equivalent to the integral multiple (record count twice) of the fixed block length of a FBA record, for example, the control information added by the following (d), so that it may be settled per 12KB.

[0026] Next, as the control information which shows the location of all the CKD records contained in it is shown in drawing 3 (d) and drawing 4 (d) to the management unit shown in drawing 3 (c), it writes to the tail end of a management unit in an order from back.

[0027] It is not necessary to secure beforehand the location which memorizes positional information, and the total capacity of positional information is to be reduced by making adjustable size of the positional information within the management unit of a record in drawing 4 (d) here. That is, a variable-length record is indicated from the head of a management unit, and it can memorize until the number of positional information and the number of variable-length records are in agreement and both storage region collides within a management unit, in order that positional information may be memorized in an order from the back of a management unit and may go. Thus, control only whose part of the record actually recorded uses a positional information field is enabled. However, although the control at the time of data transfer and a format light will be able to become complicated if size of a positional information field is made adjustable, the following methods are adopted here and the improvement is aimed at.

[0028] Now, in drawing 4 (d), the positional information of a control frame and a record is called a record pointer for the field which records a frame for the management unit of the positional information of a record, and records a record frame and control information for the field which actually records a CKD record. - Only the part of the record actually recorded uses a record pointer. Thereby, if the size of a record pointer field sets size of a management unit to 12KB, and a maximum of 94 CKD records are recorded and it sets it to 4B as positional information per record, it can be managed per 1CKD truck and with 4Bx94=276B at the maximum. The positional information of this 4B is shown as a record pointer in drawing 4 (d), and the record pointer consists of a record number of 1B, a sector value of 1B, and a memory address of 2B. The record number records the number of the variable-length record recorded. A sector value memorizes a sector value in case the record recorded is recorded by the CKD record method. A memory address memorizes the address in the frame of the record memorized by the frame. A sector value is used by being compared with the sector value specified by the set sector command. Similarly, since it is compared with the record number specified by the search ID command, a record number is used. A memory address shows the address of the actual record memorized by the record frame. When the sector value and record number of a record pointer are compared by a set sector command and the search ID command and it is in agreement,

this memory address is used and a record is read.
[0029] The size of the control frame containing a record pointer is separately held as control information, i.e., a control frame pointer. Using this information, even if the size of a control frame serves as adjustable, it is not necessary to complicate control.
[0030] The control frame which are such control information is recorded on the tail end instead of a head of each frame. Moreover, the record pointer in a control frame is also arranged in an order from back. The record pointer corresponding to it in what has an actual record in front within a record frame comes back within a control frame. Thereby, unreasonableness also of the control at the time of a format light is lost.
[0031] Next, as shown in drawing 4 (d), after [, such as positional information of a CKD record,] addition, as 12KB of record frame is shown in drawing 5 (e), it divides fairly to the block size (1KB) of a FBA record, 1024B [for example,], and as are shown in drawing 5 (f), and an ID field is prepared in the data field of each record, respectively and it is shown in drawing 5 (g), the disk of a FBA record method is obtained further.
[0032] Moreover, it leaves the information which shows the relative position from the head of the truck in a CKD format for every one CKD record currently recorded in the fixed block, for example, a segment number. If by doing in this way shows the location of each record, the track capacity on the CKD format already immediately consumed from the information which shows the relative position currently recorded on the record is known.
[0033] Next, actuation of the return disk storage control 5 is explained to drawing 2 . Disk storage control 5 has the channel pass server 51 (henceforth CPS) which performs conversion and a transfer of data between a host computer 1 and the FBA disk 9. It has the input section 52 for inputting a command and data into the channel pass server 51 from a host computer 1. Moreover, it has the output section 52 for outputting the data from the FBA disk 9 to a host computer 1. When the access instruction of the variable-length record method by the software which is operating with the host computer 1 from the input section 52 is inputted, the location count section 53 calculates the location of the management unit considered that the variable-length record accessed based on the sector value specified for example, by the set sector command is memorized. Since all gaps are removed when data are recorded on the FBA disk 9 as having mentioned above the count by this location count section, data are memorized in the form where it was got blocked forward, rather than the case where it exists in the CKD disk 10, and the location count section 53 calculates the location of the management unit considered that the variable-length record which it is going to access by guess is memorized. The record retrieval section 54 searches the record which is going to read a management unit into cache memory 6, and is going to access it based on the location of the management unit calculated by the location count section 53. The management unit retrieval section 56 compares the relative-position information on the record which it is going to access (sector value) with the relative-position information (sector value) currently held at the searched management unit, and it reads into cache memory in order the management unit which exists in the FBA disk 9 until a management unit with the relative-position information which it is going to access is found. The record specification section 57 specifies the variable-length record which it is going to access when the management unit made into the purpose by the management unit retrieval section 56 is searched using the address information contained in the management unit. The manager the data pass server (DPS) 55 performs [manager] R/W of the data between the FBA disk 9 and cache memory 6 using the FBA command performs the whole control and management of cache memory 6.
[0034] The three points become important in case CKD-FBA conversion is performed. That is, where [of the track format of I.CKD] do you leave and where delete?

II. -- after processing track format such, how is the location of the original record found?
How do you check the track capacity of III.CKD, or how detect track overrun?
It must inquire as the point of three **. Considering the three above-mentioned points, it turns out that other variations can be considered. Hereafter, a variation is considered about each point in sequence.
[0035] First, about I., after ECC, padding, etc. generate the information added by the disk unit side from the first by DKC, changing into FBA is useless. Therefore, "where where does it leave and is deleted" will say as a matter of fact, "Which gap do leave and which gap is deleted?" (what is necessary is to include in a gap the control information and the physical parameter which had been managed by DKC(s) like defect information other than ECC or padding, and just to consider them). This gap leaves and the following variations also including the conventional method can be considered about the direction (refer to drawing 6).
< case I-1> ... All gaps also including the approach and inter-record gap of "IBM4321 / 4331 processor compatibility facility" will be deleted.
< case I-2> ... It leaves only the conventional approach and inter-record gap, and the gap between the fields is removed.
< case It leaves I-3> and not only an inter-record gap but the gap between all the fields (ECC, padding, etc. remove).
[0036] Still more nearly various variations can be considered by what these all directions methods carry out treatment of fine information other than a gap, respectively. Moreover, although these all directions methods must be combined with a certain means (the positioning approach of a record) of II, they can consider some following approaches also including the conventional approach to this (referring to drawing 7 , and each case and one to one of I are not necessarily supported).
< case It has no special information which shows the location of II-1> and a record.
- When looking for a record, always read in an order from the record of the head of a truck, and calculate the location of the count field of the following record from the value of KL and DL which are contained in the count field of each record.
< case II-2> ... The conventional approach and the information which shows the location of a record are partially held apart from the record itself. (By the conventional approach, the location of the count field of the record located in the head of each FBA block is held)
- From the record in which a location is shown, read a back record like <case II-1> in an order from the record which the location understands, and it calculates the location of the following record from KL of each record, and DL.
< case The information which shows the location of the record of II-3> and all is held apart from the record itself.
- All the record looks for a location from this information.
- the positional information of a record can consider how (for example, head of a truck) to bundle up a part for all records and hold to one place, and the approach (for example, every -- the information which shows the location of the record contained in the FBA block at the head of a FBA block is held) of holding dispersedly.
< case Also on the logical truck image after II-4> and FBA conversion, a certain address mark (AM) is generated and a record is looked for for this to a mark (AM search is performed also on memory). However, by raising a memory load superfluously, in order that this approach may perform AM search on having not invented the method of generating the nice address mark concretely, and memory, since it is inefficient-like, it excepts reading all data in order from the range of examination.

[0037] Subsequently, the following approaches can be considered about the check approach of the track capacity of III (refer to drawing 8). These approaches have what the gap of I leaves and can be applied depending on the direction, and the thing which is not made.
< case KL of all the records from the head of III-1> and a truck and DL are got to know, and the track capacity on the CKD format already consumed is calculated by all carrying out the guide peg also of the gap of a between, or the part (as for these all, the size having been decided uniquely) of padding.
< case III-2> ... The die length of the gap (the conventional approach inter-record gap) which remains the part of a conventional approach and conventional eliminated ECC, or padding is lengthened, and it is made consistent. This maintains the relative position from the head of the truck of each record with the original CKD truck.
- Therefore, if the location of each record is known, the track capacity on the CKD format already immediately consumed from the address on the memory of the record is known.
< case It leaves the information (for example, segment number) which shows the relative position from the head of the truck in a CKD format to III-3> and each record.
- Therefore, if the location of each record is known, the track capacity on the CKD format already immediately consumed from the information which shows the relative position currently recorded on the record is known.
[0038] At drawing 8 , it is < case. III-1> and < case III-3> is a case < case which has removed all gaps. Although expressed by I-1>, as for each of these approaches, the gap may remain how. Moreover, < case III-2> is the case < case where it has left only the inter-record gap. Although expressed by I-2>, the case <a case I-3> where it leaves all gaps is sufficient. However, < case III-2> is a < case which removes all gaps. In I-1>, since a memory address is unmaintainable with accommodation of gap length with the relative position of the original record, implementation is impossible on a principle. Some (there being also a combination impossible as mentioned above and a meaningless combination) CKD-FBA conversion methods can be done with the combination of each of these cases of I, II, and III. The conventional method is < case. I-2>, < case II-2>, < case It can be called the CKD-FBA conversion method by the combination of III-2>.
[0039] The list of effective things is shown in drawing 9 among the combination of each above-mentioned case. The unrealizable thing among the combination of each case and the meaningless thing are removed beforehand. Explanation of the column of a compare item is shown after the following knot. The following combination mentioned above is unrealizable.
- < case I-1> and < case Combination < case of III-2> I-1> is < case where a memory address is maintained by accommodation of gap length with the relative position of the original record since it is the method which removes all gaps. The check approach of the track capacity of III-2> is impossible. Moreover, although the following combination is not impossible, it is meaningless on a method by the following reasons.
- < case II-2> and < case Combination -< case of III-1> II-3> and < case Combination < case of III-1> II-2> and < case The record of each of II-3> is a method which discovers a record from the positional information of the record currently held independently. Therefore, there is an advantage that it is not necessary to read to looking for a certain record in an order from the record of the beginning on a truck. However, < case In order to check track capacity, the data of all the count fields on a truck are required for III-1>. Therefore, in such combination, it will read in an order from the record of the beginning on a truck at once after all, and is < case. II-2> or < case The semantics which adopted the method of II-3> is lost. Namely, < case III-1> is a < case which reads all the records on a truck as a positioning method of a

record. Only when II-1> is adopted, it can be called a track capacity check method with semantics. All of combination other than these are put on drawing 9 . The party shows the CKD-FBA conversion method realized in one combination, and the view of drawing 9 is a method with which the method which attaches O is adopted in the combination. As a method of how [therefore,] of for example, a party eye to leave "gap, as for a method", it is < case. I-1> is adopted and it is < case about the "record positioning approach". Considering II-1> as "the check approach of track capacity", it is < case. It will be called the CKD-FBA conversion method which adopted III-1>. From now on, <1, 1, 1> will be written for convenience. [such a method] In this notation, the conventional method can be expressed as <2, 2, 2>.

[0040] Henceforth, although each CKD-FBA conversion method is compared, the comparison point of each conversion method is first considered as follows to eye others.

a. Size of memory space at the time of developing the truck image of a CKD format on the cache memory (only henceforth memory) of disk storage control (DKC). ... This memory space cannot but take not only a conversion method but more capacity for one truck in accordance with the worst value after all, in order to be greatly dependent on the track format at that time (size of record count etc.). And since this memory image will be written to the physical disk of FBA, it becomes the integral multiple of the block (sector) capacity of a FBA disk. Therefore, a little size is not a not much serious problem. It is based mainly on the size of additional information required for positioning of how to leave a gap and a record etc.

b. The ease of doing of data transfer ... Having continued also on memory tends to do the part which carries out data transfer continuously between channels. For example, having continued also on memory tends to transmit key field and a data field. However, about the part in which the gap existed from the first, since the direction of a channel may cause overrun if time amount is not taken, it is seldom necessary to hurry by the time thing.

c. Procedure when finding the purpose record ... It is a procedure for finding the purpose record within the truck image of the CKD format on memory. Moreover, it is necessary to only take into consideration the numerousness of the counts of memory access not only the count of a procedure therefore but required etc. at this time. Since a truck image is developed on the cache memory which is a common area, even if there are a little many procedures (for example, computational complexity) itself, the direction with few counts of memory access is advantageous in the whole system. This count of memory access is mainly decided by the positioning approach of a record.

d. Procedure at the time of a format light ... It is necessary to investigate the adjustment of KL of the already formatted record, KL of the record with which it has been sent by ** track capacity and the host on memory at the time of a format light although what is necessary is just to carry out DL **** light, and DL at the time of an update light. That is, the check of track overrun is required. Furthermore, since it is necessary to maintain control information other than a record depending on a method, it is necessary to take into consideration the procedure for it, the numerousness of memory accesses, etc.

Hereafter, each description of each CKD-FBA conversion method and fine analysis are performed based on this comparison point.

[0041] <1, 1, 1> and gap: -- all -- from the record of the beginning of deletion and a record positioning:truck -- all -- since the record of the beginning of a record lead track capacity check:truck to not all record lead a. memory space this gentleman types also have all gaps and there is also no addition of control information, the use effectiveness of memory is the best.

b. What is necessary is just to count up the memory address, when the field which continued between channels carries out data transfer, since each field is continuing on memory in the ease of doing of data transfer.

c. In order to find the procedure purpose record when finding the purpose record, only the record count from the head of a truck to an applicable record needs to access memory. That is, the lead of the count field of each record is needed. After acquiring the count field of each record, from those KL(s) and DL, calculating the location of the purpose record adds **** without an excessive gap etc., and KL and DL, and it is [what is necessary] to just be crowded and is simple.
d. Also in order to check the procedure track capacity at the time of a format light, only the record count from the head of a truck to an applicable record needs to access memory. It is necessary to also convert parts, such as a deleted gap, conversely for the check of track capacity. There is no control information which must be updated at the time of a format light.
[0042] <1, 2, 3> and a gap : although all maintenance a. memory space gaps delete relative-position information, such as a segment number, on maintenance and track capacity check:each record, the location of the first record altogether contained in deletion and a record positioning:FBA block Only the part of the control information for holding the location of the first record contained in the block of FBA, the segment number of each record, etc. uses memory for an excess from <1, 1, 1>.
b. What is necessary is just to count up the memory address, when the field which continued between channels carries out data transfer, since each field is continuing on memory too in the ease of doing of data transfer. However, since it is divided by the positional information of the first record contained in a FBA block also on memory etc. when the block of FBA is straddled even if it is in the single field, it is necessary to fly the part of the positional information like a skip defect, and to perform data transfer.
c. In order to look for the procedure purpose record when finding the purpose record, only the count of the record which exists in from the record of the beginning of a block of FBA before the purpose record needs to access memory. Therefore, there are few counts of memory access than <1, 1, 1>. Most approaches of calculating the location of the purpose record from the location of the record of the beginning of a block of FBA are the same as <1, 1, 1>, and simple (what is necessary is just to add the size of the count field, KL, and DL).
d. from the segment number (information which shows the relative position on the truck of the record in an original CKD format) of the record in front of [of the record which is going to carry out the format light for the check of the procedure track capacity at the time of a format light] one, KL, and DL -- back -- calculate whether the light of the record of the size of which can be carried out. When a format light is carried out, the segment number of the record must be written to the record itself which carried out the light. When the new count field is generated on a new FBA block, the information which shows the location of the count field of a top record to the FBA block must be written to coincidence.
[0043] <1, 3, 3>, and gap: -- all -- deletion and record positioning: -- there is still more memory space than the case where only the location of the first record contained in the block of FBA as shown in <1, 2, 3> is held in order to hold the positional information of all records independently although all maintenance a. memory-space gaps delete relative-position information, such as a segment number, for the relative-position information on all records on maintenance and track capacity check:each record, and it is the need. It is the same as <1, 2, 3> to hold a segment number etc. on each record.
b. It is the same as <1, 2, 3> almost in the ease of doing of data transfer. What is necessary is just to count up the memory address, when the field which continued between channels carries out data transfer, since each field is continuing on memory. Since it is divided by the positional information of the record contained in a FBA block also on memory etc. when straddling the block of FBA even if it is in the single field, when it distributes and has the positional information of a record for every block of FBA, it is necessary to fly the part of the positional information like a skip defect, and to perform data transfer.

c. Since what is necessary is to read the positional information of a record only once in order to look for the procedure purpose record when finding the purpose record, the count of memory access is min in all methods. The information shows the location of the purpose record direct.
d. Since the check approach of the procedure at the time of a format light <1, 2, 3> and track capacity is the same, the procedure at the time of a format light is almost the same as <1, 2, 3>. from the segment number (information which shows the relative position on the truck of the record in an original CKD format) of the record in front of [of the record which is going to carry out the format light for the check of track capacity] one, KL, and DL -- back -- it calculates whether the light of the record of the size of which can be carried out. When a format light is carried out, the segment number of the record must be calculated and written to the record itself which carried out the light. The information which shows the location of the count field of the record must be added to the control information which surely holds the positional information of a record to coincidence.
[0044] from the record of the beginning of the - record positioning:truck with which deletion and an inter-record gap leave only <2, 1, 1> and the gap between the gap:fields -- all -- all the record lead a. memory space from the record of the beginning of a record lead track capacity check:truck -- although there is no addition of excessive control information, since it leaves an inter-record gap, memory space is the need more mostly than the <1, *, *> method. However, since it is not maintaining the relative position of a record by the inter-record gap which it left, it is not necessary to lengthen an inter-record gap by force. Therefore, although memory space is made few compared with <*, *, 2> method which lengthens inter-record gap length and maintains the relative position of a record as the check approach of track capacity, when it says conversely, there is also no semantics which left the inter-record gap.
b. What is necessary is just to count up a memory address, when transmitting the data of the field which continued between channels (C->K->D), since there is no gap between the fields although an inter-record gap exists in the ease of doing of data transfer. Since there is also no excessive control information, even if the single field straddles the block of FBA, the field is not divided on memory.
c. It is almost the same as the procedure <1, 1, 1> when finding the purpose record. In order to find the purpose record, only the record count from the head of a truck to an applicable record needs to access memory. That is, the lead of the count field of each record is needed. After acquiring the count field of each record, in order to calculate the location of the purpose record from those KL(s) and DL, in addition to KL and DL, an inter-record gap is added, and it is [what is necessary] to just be crowded and is simple.
d. the procedure at the time of a format light -- this is almost the same as <1, 1, 1>. Also in order to check track capacity, only the record count from the head of a truck to an applicable record needs to access memory. It is necessary to also convert parts, such as a deleted gap between the fields, for the check of track capacity. There is no control information which must be updated at the time of a format light.
[0045] <2, 2, 2> Deletion of only the gap between the fields ... a method and conventional gap: -- - record positioning which an inter-record gap leaves : by accommodation of maintenance and track capacity check:gap length, although the gap between the maintenance a. memory space fields etc. deletes the relative position of each record, the first record location included in the block of FBA Since the part inter-record gap is lengthened and the relative position of each record is maintained with the original CKD format, memory space required after all It is the same as the track capacity of the original CKD format, and memory is too much required only for the part of the positional information of the first record further contained in the block of FBA.
b. What is necessary is just to count up a memory address, when transmitting the data of the field which continued between channels (C->K->D), since there is no gap between the fields although an inter-record

gap exists in the ease of doing of data transfer. However, since it is divided by the positional information of the first record contained in a FBA block also on memory etc. when the block of FBA is straddled unlike <2, 1, 1>, even if it is in the single field, it is necessary to fly the part of the positional information like a skip defect, and to perform data transfer.
c. In order to look for the procedure purpose record when finding the purpose record, only the count of the record which exists in from the record of the beginning of a block of FBA before the purpose record needs to access memory. Therefore, there are few counts of memory access than <2, 1, 1>. Most approaches of calculating the location of the purpose record from the location of the record of the beginning of a block of FBA are the same as <2, 1, 1>, and simple (what is necessary is just to add an inter-record gap to the size of the count field, KL, and DL pan rapidly). Moreover, since the relative position in a CKD format of each record is maintained also on memory, it is Set. The block of FBA with which the purpose record is contained from the value of Sector can be searched for correctly.
d. if the address on the memory of the record which is going to carry out the format light to it since the relative position in a CKD format of each record is maintained on the check of the procedure track capacity at the time of a format light also on memory is known (it understands in the procedure of c.) -- immediately -- back -- it is calculable whether the light of the record of the size of which can be carried out. When a format light is carried out and the new count field is generated on a new FBA block, the information which shows the location of the count field of a top record must be written to the FBA block.
[0046] Only the gap between the fields <2, 2, 3> and a gap : Deletion, - record positioning which an inter-record gap leaves : The gap between the maintenance a. memory space fields etc. deletes relative-position information, such as a segment number, for the first record location included in the block of FBA on maintenance and track capacity check:each record. Although it is the same as <2, 2, 2> to leave an inter-record gap, since it is not maintaining the relative position of a record by the inter-record gap which it left, it is not necessary to lengthen an inter-record gap by force (therefore, the semantics which leaves a gap too becomes thin). However, the part of the positional information of the first record contained in the block of FBA is required for an excess. Moreover, since the part of the segment number contained in each record is also required, as memory space, and it becomes less than <2, 2, 2> after all than <2, 1, 1>.
b. This is completely the same as <2, 2, 2> in the ease of doing of data transfer. What is necessary is just to count up a memory address, when transmitting the data of the field which continued between channels (C->K->D), since there is no gap between the fields although an inter-record gap exists (since the segment number currently held at each record is contained in the count field, it does not become the obstacle of C->K at the time of data transfer). However, since it is divided by control information, such as positional information of the first record contained in a FBA block also on memory, when the block of FBA is straddled even if it is in the single field, it is necessary to fly the part of the control information like a skip defect, and to perform data transfer.
c. the procedure when finding the purpose record -- this is completely the same as <2, 2, 2>. In order to look for the purpose record, only the count of the record which exists in from the record of the beginning of a block of FBA before the purpose record needs to access memory. Therefore, there are few counts of memory access than <1, 1, 1>. Most approaches of calculating the location of the purpose record from the location of the record of the beginning of a block of FBA are the same as <1, 1, 1>, and simple (what is necessary is just to add an inter-record gap to the size of the count field, KL, and DL pan rapidly).
d. from the segment number (information which shows the relative position on the truck of the record in an original CKD format) of the record in front of [of the record which is going to carry out the format light for the check of the procedure track capacity at the time of a format light] one, KL, and DL -- back -- calculate

whether the light of the record of the size of which can be carried out. When a format light is carried out, the segment number of the record must be calculated and written to the record itself which carried out the light. When the new count field is generated on a new FBA block, the information which shows the location of the count field of a top record to the FBA block must be written to coincidence.

[0047] Only the gap between the fields <2, 3, 2> and a gap : Deletion, - record positioning: which an inter-record gap leaves -- the relative-position information on all records -- maintenance and track capacity check: -- by accommodation of gap length Since the relative position of each record is maintained with the original CKD format of the relative position of each record by accommodation of maintenance a. memory space ****** inter-record gap length Required memory space is fundamentally the same as the track capacity of the original CKD format like <2, 2, 2>. However, since the positional information of the record added as control information holds the part of not only the part of the record of the beginning of a block of FBA but all records as shown in <2, 2, 2>, its memory space also increases more than <2, 2, 2>, and it is all the maxes in a method.

b. It is completely the same as <2, 2, 2>, and <2, 2, 3> in the ease of doing of data transfer. What is necessary is just to count up a memory address, when transmitting the data of the field which continued between channels (C->K->D), since there is no gap between the fields although an inter-record gap exists. However, since it is divided by the positional information of the record contained in a FBA block also on memory etc. when the block of FBA is straddled even if it is in the single field, when it distributes and has the positional information of a record for every block of FBA, it is necessary to fly the part of the positional information like a skip defect, and to perform data transfer.

c. It is completely the same as the procedure <1, 3, 3> when finding the purpose record. Since what is necessary is to read the positional information of a record only once in order to look for the purpose record, the count of memory access is min in all methods. The information shows the location of the purpose record direct. Moreover, since the relative position in a CKD format of each record is maintained also on memory, it is Set. The block of FBA with which the purpose record is contained from the value of Sector can be searched for correctly.

d. It becomes a procedure at the time of a format light <2, 2, 2>, and the in-between procedure of <1, 3, 3>. Since the relative position in a CKD format of each record is maintained also on memory on the check of track capacity, if the address on the memory of the record which is going to carry out the format light is known (it understands in the procedure of c.), on it, it is calculable whether the light of the record of the size of which can be carried out the back immediately. When a format light is carried out, the information which shows the location of the count field of the record must be added to the control information which surely holds the positional information of a record.

[0048] Only the gap between the fields <2, 3, 3> and a gap : Deletion, The gap between the maintenance a. memory space fields etc. deletes relative-position information, such as a segment number, on each record. - record positioning: which an inter-record gap leaves -- the relative-position information on all records -- maintenance and track capacity check: -- Although it is the same as <2, 3, 2> to leave an inter-record gap, since it is not maintaining the relative position of a record by the inter-record gap which it left, it is not necessary to lengthen an inter-record gap by force (therefore, the semantics which leaves a gap too becomes thin). However, memory space is too much required for the part of the positional information of all records. Moreover, since the part of the segment number contained in each record is also required, and it becomes less than <2, 3, 2> than <2, 2, 3> which are the method which was well alike as memory space after all.

b. This is completely the same as <2, 3, 2> in the ease of doing of data transfer. What is necessary is just

to count up a memory address, when transmitting the data of the field which continued between channels (C->K->D), since there is no gap between the fields although an inter-record gap exists (since the segment number currently held at each record is contained in the count field, it does not become the obstacle of C->K at the time of data transfer). Moreover, since it is divided by the positional information of the record contained in a FBA block also on memory etc. when straddling the block of FBA even if it is in the single field, when it distributes and has the positional information of a record for every block of FBA, it is necessary to carry out the part of the positional information like a skip defect, to fly it, and to perform data transfer.
c. the procedure when finding the purpose record -- this is completely the same as <2, 3, 2>. Since what is necessary is to read the positional information of a record only once in order to look for the purpose record, the count of memory access is min in all methods. The information shows the location of the purpose record direct.
d. from the segment number (information which shows the relative position on the truck of the record in an original CKD format) of the record in front of [of the record which is going to carry out the format light for the check of the procedure track capacity at the time of a format light] one, KL, and DL -- back -- calculate whether the light of the record of the size of which can be carried out. When a format light is carried out, the segment number of the record must be calculated and written to the record itself which carried out the light. The information which shows the location of the count field of the record must be added to the control information which surely holds the positional information of a record to coincidence.
[0049] from the record of the beginning of the - record positioning:truck which leaves <3, 1, 1> and a gap:inter-record gap, and the gap between the fields -- all -- all the record lead a. memory space from the record of the beginning of a record lead track capacity check:truck -- although there is no addition of excessive control information, since it leaves an inter-record gap and the gap between the fields, and the truck image of a CKD format is left behind mostly as it is, memory space becomes close to max. However, since it does not say that the relative position of the record in a CKD format will be maintained positively, the part of ECC or padding does not need to add to memory space. Instead, the semantics which left the gap is lost not much.
b. Since a gap exists also on memory in the ease of doing of data transfer between records and as for between the fields, when performing data transfer of the continuous field between channels, it not only counts up a memory address simply, but it needs to fly and reset the part of a gap. However, since there is no excessive control information, even if the single field straddles the block of FBA, the field is not divided on memory.
c. It is almost the same as the procedure <1, 1, 1> when finding the purpose record. In order to find the purpose record, only the record count from the head of a truck to an applicable record needs to access memory. That is, the lead of the count field of each record is needed. After acquiring the count field of each record, in order to calculate the location of the purpose record from those KL(s) and DL, in addition to KL and DL, an inter-record gap and the gap between the fields are added, and it is [what is necessary] to just be crowded and is simple.
d. the procedure at the time of a format light -- this is almost the same as <1, 1, 1>. Also in order to check track capacity, only the record count from the head of a truck to an applicable record needs to access memory. Although the gap remains as it is for the check of track capacity, parts, such as ECC and padding data, need to convert. Although ** track capacity is immediately known from a memory address if the same capacity is exactly amended by gap length including the part of ECC etc., this serves as <3, 2, 2> method. There is no control information which must be updated at the time of a format light.
[0050] <3, 2, 2> and a gap : An inter-record gap, The relative position of each record by accommodation of

maintenance and track capacity check:gap length for the location of the first record contained in the block of FBA - record positioning which leaves the gap between the fields : A maintenance a. memory space inter-record gap, Although the gap between the fields remains Since gap length is adjusted and the relative position of each record is maintained with the original CKD format Memory space required after all is the same as <2, 2, 2>, and only the part which added the positional information of the first record contained in the track capacity of the original CKD format and the block of FBA is required for it.
b. Since a gap exists also on memory in the ease of doing of data transfer between records and as for between the fields, when performing data transfer of the continuous field between channels, it not only counts up a memory address simply, but it needs to fly and reset the part of a gap like <3, 1, 1>. Furthermore, since it is divided by the positional information of the first record contained in a FBA block also on memory etc. when the block of FBA is straddled unlike <3, 1, 1>, even if it is in the single field, it is necessary to fly the part of the positional information like a skip defect, and to perform data transfer.
c. It is completely the same as the procedure <2, 2, 2> when finding the purpose record. In order to look for the purpose record, only the count of the record which exists in from the record of the beginning of a block of FBA before the purpose record needs to access memory. Therefore, there are few counts of memory access than <3, 1, 1>. Most approaches of calculating the location of the purpose record from the location of the record of the beginning of a block of FBA are the same as <3, 1, 1>, and simple (what is necessary is just to add an inter-record gap and the gap between the fields to the size of the count field, KL, and DL pan rapidly). Moreover, since the relative position in a CKD format of each record is maintained also on memory, it is Set. The block of FBA with which the purpose record is contained from the value of Sector can be searched for correctly.
d. the procedure at the time of a format light -- this is completely the same as <2, 2, 2>. Since the relative position in a CKD format of each record is maintained also on memory on the check of track capacity, if the address on the memory of the record which is going to carry out the format light is known (it understands in the procedure of c.), on it, it is calculable whether the light of the record of the size of which can be carried out the back immediately. When a format light is carried out and the new count field is generated on a new FBA block, the information which shows the location of the count field of a top record must be written to the FBA block.
[0051] <3, 2, 3> and a gap : An inter-record gap, Relative-position information, such as a segment number, on maintenance and track capacity check:each record for the location of the first record contained in the block of FBA - record positioning which leaves the gap between the fields : A maintenance a. memory space inter-record gap, Since it leaves the gap between the fields, mostly, the truck image of a CKD format is left behind as it is, and since the part of the positional information of the first record contained in the block of FBA is also required for an excess, memory space becomes close to max further. However, since it does not say that the relative position of the record in a CKD format will be maintained positively, the part of ECC or padding does not need to add to memory space. Instead, the semantics which left the gap not much is lost.
b. This is completely the same as <3, 2, 2> in the ease of doing of data transfer. It is necessary to fly the part of a gap and to reset on memory as well as not only counting up a memory address simply, when performing between channels data transfer of the field which continued since the gap existed between records and as for between the fields but <3, 1, 1>. Furthermore, since it is divided by the positional information of the first record contained in a FBA block also on memory etc. when the block of FBA is straddled unlike <3, 1, 1>, even if it is in the single field, it is necessary to fly the part of the positional information like a skip defect, and to perform data transfer.

c. the procedure when finding the purpose record -- this is completely the same as <3, 2, 2>. In order to look for the purpose record, only the count of the record which exists in from the record of the beginning of a block of FBA before the purpose record needs to access memory. Therefore, there are few counts of memory access than <3, 1, 1>. Most approaches of calculating the location of the purpose record from the location of the record of the beginning of a block of FBA are the same as <3, 1, 1>, and simple (what is necessary is just to add an inter-record gap and the gap between the fields to the size of the count field, KL, and DL pan rapidly).
d. from the segment number (information which shows the relative position on the truck of the record in an original CKD format) of the record in front of [of the record which is going to carry out the format light for the check of the procedure track capacity at the time of a format light] one, KL, and DL -- back -- calculate whether the light of the record of the size of which can be carried out. When a format light is carried out, the segment number of the record must be written to the record itself which carried out the light. When the new count field is generated on a FBA block new to coincidence, the information which shows the location of the count field of a top record must be written to the FBA block.
[0052] <3, 3, 2> and a gap : An inter-record gap, - record positioning: which leaves the gap between the fields -- the relative-position information on all records -- maintenance and track capacity check: -- by accommodation of gap length Since the relative position of each record is maintained with the original CKD format of the relative position of each record by accommodation of maintenance a. memory space ****** inter-record gap length Required memory space is fundamentally the same as the track capacity of the original CKD format like <3, 2, 2>. However, since the positional information of the record added as control information holds the part of not only the part of the record of the beginning of a block of FBA but all records as shown in <3, 2, 2>, its memory space also increases more than <3, 2, 2>, and it is all the maxes in a method like <2, 3, 2>.
b. This is completely the same as <3, 2, 2>, and <3, 2, 3> in the ease of doing of data transfer. It is necessary to fly the part of a gap and to reset on memory as well as not only counting up a memory address simply, when performing between channels data transfer of the field which continued since the gap existed between records and as for between the fields but <3, 1, 1>. Furthermore, since it is divided by the positional information of the record contained in a FBA block also on memory etc. when the block of FBA is straddled even if it is in the single field, when it distributes and has the positional information of a record for every block of FBA unlike <3, 1, 1>, it is necessary to fly the part of the positional information like a skip defect, and to perform data transfer.
c. It is completely the same as the procedure <1, 3, 3> when finding the purpose record, <2, 3, 3>, etc. Since what is necessary is to read the positional information of a record only once in order to look for the purpose record, the count of memory access is min in all methods. The information shows the location of the purpose record direct. Moreover, since the relative position in a CKD format of each record is maintained also on memory, it is Set. The block of FBA with which the purpose record is contained from the value of Sector can be searched for correctly.
d. It is completely the same as the procedure at the time of a format light <2, 3, 2>. Since the relative position in a CKD format of each record is maintained also on memory on the check of track capacity, if the address on the memory of the record which is going to carry out the format light is known (it understands in the procedure of c.), on it, it is calculable whether the light of the record of the size of which can be carried out the back immediately. When a format light is carried out, the information which shows the location of the count field of the record must be added to the control information which surely holds the positional information of a record.

[0053] Relative-position information, such as a segment number, on maintenance and track capacity check:each record for the relative-position information on all records - record positioning which leaves <3, 3, 3> and a gap:inter-record gap, and the gap between the fields : A maintenance a. memory space inter-record gap, Since it leaves the gap between the fields, the truck image of a CKD format is left behind mostly as it is, and since the part of the positional information of all records is also still more nearly required for an excess, from <3, 2, 3>, there is still more memory space and it becomes close to max. However, since it does not say that the relative position of the record in a CKD format will be maintained positively, the part of ECC or padding does not need to add to memory space. Instead, the semantics which left the gap is lost not much.

b. This is completely the same as <3, 2, 2>, <3, 2, 3>, and <3, 3, 2> in the ease of doing of data transfer. It is necessary to fly the part of a gap and to reset on memory as well as not only counting up a memory address simply, when performing between channels data transfer of the field which continued since the gap existed between records and as for between the fields but <3, 1, 1>. Furthermore, since it is divided by the positional information of the record contained in a FBA block also on memory etc. when the block of FBA is straddled even if it is in the single field, when it distributes and has the positional information of a record for every block of FBA unlike <3, 1, 1>, it is necessary to fly the part of the positional information like a skip defect, and to perform data transfer.

c. the procedure when finding the purpose record -- this is completely the same as <3, 3, 2>. Since what is necessary is to read the positional information of a record only once in order to look for the purpose record, the count of memory access is min in all methods. The information shows the location of the purpose record direct.

d. It is completely the same as the procedure at the time of a format light <2, 3, 3>. from the segment number (information which shows the relative position on the truck of the record in an original CKD format) of the record in front of [of the record which is going to carry out the format light for the check of track capacity] one, KL, and DL -- back -- it calculates whether the light of the record of the size of which can be carried out. When a format light is carried out, the segment number of the record must be written to the record itself which surely carried out the light. The information which shows the location of the count field of the record must be added to the control information which holds the positional information of a record to coincidence.

[0054] The comparison result of an all directions type mentioned above in the right half of drawing 9 was summarized. Four compare items mentioned above are indicated in the column of a "compare item." According to the significance, five steps of grade attachments are carried out to each compare item. 5 is the most important item and 1 is an item which is not not much important. Moreover, in each of this compare item, five-step evaluation was performed and the mark of one to five points are given to each conversion method. Five points were attached to the best conversion method about a certain compare item, and one point is given to the most inferior conversion method. Thus, the mark gave are totaled for every conversion method, and it is considering as the sum total score of the conversion method. The approach of a total has totaled what applied the value of the grade of the compare item to the prime spot of each compare item about all compare items. Therefore, it will be called the conversion method which was as excellent as the high conversion method of a sum total score. In drawing 9 , this sum total score gives * to the sum total score column of a conversion method higher than the conventional method.

[0055] The sum total score of drawing 9 also has a much subjective field, and is a standard to the last. However, even if it shook some values of the grade of how to give mark to a trial or each compare item, the conversion method which becomes the conventional method and the score more than equivalent was the

same as drawing 9 too. Therefore, as long as it thinks according to this compare item, it is thought that it becomes a good standard. The reason to which the conventional method was not topping is the count of the memory access in the procedure when mainly finding the purpose record. In the conventional method, although the record of the beginning under block of FBA can be found by one shot, the record behind that must find and read the count field of a consecutive record in sequence from there. The reason the conventional method did not think this point as important can consider the two following points.

(1) The method of the problem former of a unit with the positional information of a record supposes that it has the positional information of the first record contained in every [of a FBA disk] block (** sector) there. The size of the record of CKD recorded there by the actual SCSI disk to the magnitude of this block being about at most 1-2KB has most about 4KB. Therefore, it is thought that the CKD record with which one CKD record is actually contained in one FBA block after all ranging over two or more FBA blocks like drawing 10 rather than the case where many CKD records to one FBA block are contained has more cases of one or less piece. Therefore, Set If Sector shows exactly even the FBA block with which the target record is contained, since the purpose record can be found by one memory access, the count of memory access will hardly increase also by this method, either.

(2) In the method of the problem former of the place for a track buffer, although the emulator which performs such CKD-FBA conversion may be anywhere between a channel processor and a FBA disk (DKC will be sufficient, of course), the channel processor itself is performing the emulator. That is, he is seldom conscious of the large-scale system which used the disk cache as the base. Therefore, the memory for track buffers which develops a truck image also considers local memory of a channel processor the primary storage of the host CPU, and own. Thus, especially a track buffer is in own local memory of a channel processor, and when the channel processor performs a CKD-FBA emulation, even if there are a little many counts of memory access there, it seldom becomes a problem.

[0056] Although it is thought that the point of the count of memory access was seldom conventionally thought as important from these points, the two above-mentioned points have conceived the following problems, respectively.

(1) It is inefficient-like although the method of the problem former of a unit with the positional information of a record has the positional information of the first record contained in every [of a FBA disk] block (** sector) there. As shown in drawing 11 , when considering an array mold disk condition (RAID) with one set of four sets of data disks, and a redundancy disk which used the disk cache as the base, the unit of the read/write to a physical disk is a part for not a fine unit like the block of FBA but 1/4 which divided a part for one truck of CKD (for example, 48KB) into each data disk truck. Therefore, it is better to also have the positional information of a record every 1/4 truck of this from every FBA (refer to drawing 11). If it has positional information every 1/4 truck, since the positional information for one truck will be collected by four places, it also ends by these four places that a CKD record is divided by this positional information on memory. However, supposing it has the positional information of a record for every block of FBA like the conventional method, the fragmentation part of one truck will also become the number of the FBA block included on one truck. (1 block 2KB -- ** -- if it carries out -- one truck -- about 24 places) that is, if it is 4KB of CKD record, 1-2 places will surely be divided among 1 record. This fragmentation part will be processed with the firmware (F/W) of disk storage control, unless special hardware (H/W) is prepared. Therefore, it supposes that it starts dozens ofmicrosec, and since the time amount which transmitting 4.5 MB/sec, then 4KB of record takes in the transfer rate of a channel is 889microsec, it leads to engine-performance [several % - about 10% per one record of] down. If positional information will be given every 1/4 truck like drawing 11 in order to avoid this, since the CKD record count contained in it will increase

shortly, the count of memory access will increase by the conventional method.
[0057] (2) Like the method of the problem former of the place for a track buffer, a track buffer is in own local memory of a channel processor, and when the channel processor performs a CKD-FBA emulation, even if there are a little many counts of memory access there, it seldom becomes a problem. However, naturally in the large-scale system considered on the assumption that a disk cache, a track buffer also comes on cache memory. Carrying out [and] data transfer of the truck image on cache memory between channels, transmitting to another track buffer becomes a big overhead in itself. On the other hand, since the emulation of CKD-FBA must be carried out to the real-time one in data transfer with a channel, the channel pass server (CPS) in DKC which manages data transfer with each channel will perform an emulation. Therefore, by the conventional method, in order to find the purpose record, each channel pass server will access frequently the track buffer on the cache memory which is a common area. Access to this cache memory is performed not only from a channel pass server but from a device side. That is, this cache memory is the bottle neck part of the system by which read/write is performed at high speed. Accessing such a part frequently in a fine unit becomes the cause which comes out so much and makes data transfer capacity of an internal bus low remarkably. Therefore, even if the amount of data of access to this cache memory increases a little, its method accessed at once is better. It is advantageous < case for the method conventional in that part not to become the lower top about mark a little, since it is above and the point of the count of memory access is also noted, but to get the high score from the conventional method in respect of this count of memory access. It is a thing in the conversion method which has adopted II-3>.
[0058] Next, "the deletion approach of a gap and evaluation of memory space" are explained. Here, in case the truck image of a CKD format is developed on cache memory (track buffer) and it records on a FBA disk, it evaluates how many memory space differ concretely, respectively about three methods mentioned above about whether it develops on memory also including a gap.
[0059] About whether in case the truck image of a CKD format is developed on cache memory (track buffer) and it records on a FBA disk, it develops on memory also including a gap, it is < case. All gaps also including I-1> and an inter-record gap will be deleted.
< case It leaves only I-2> and an inter-record gap, and the gap between the fields is removed.
< case It leaves I-3> and not only an inter-record gap but the gap between all the fields (ECC, padding, etc. remove).
Three kinds of methods to say can be considered. Here, memory space (= the block count of a FBA disk) required to develop the truck image for CKD format 1 truck for every all directions types of these is estimated and examined [measure and]. However, < case I-2>, < case About I-3> < case III-2> mentioned above as a check method of track capacity (only the part of the eliminated gap between the fields, ECC, and padding lengthens an inter-record gap) When it combines with the method which maintains the relative position from the head of the truck of each record with the original CKD truck, the memory space of a truck image always becomes equal to the original CKD format. Therefore, it estimates about the case where adjustment of gap length is not performed here.
[0060] In a CKD format, the number of gaps needs to specify track format, such as record count in the truck which estimates, in order to be dependent on the record count in 1 truck, the existence of key field, etc. and to estimate the size of the memory space by the existence of a gap. Here, as an extreme case, when the fewest [there is most record count, and], it estimates about four cases of drawing 12 , such as a standard case. All assume drawing 33 as a CKD format. Count of the record count per truck in each case is shown in drawing 13 , drawing 14 , drawing 15 , and drawing 16 . The formats 2 and 3 of drawing 12 were defined based on the investigation "work load type investigation of disk access", although not stated here.

Moreover, although the record size of a journal file is 200B-32KB, since the size used best is about 200B-300B, 200B is used by drawing 12 .
[0061] In estimating memory space, it estimates on the following conditions here.
(1) Use drawing 33 as a track format.
(2) 0 padding for uniting with ECC of each field, a tooth space (X'FF'), and 32B boundaries deletes.
(3) Leave all the parameters of others of HA and the count field.
(4) Don't estimate the capacity of control information for CKD-FBA conversion, such as positional information of a record, here. Since it is conditions different from the deletion approach of a gap again, this is estimated separately.
After all, on memory, the contents of each field other than a gap estimate noting that they become like the part shown by X of drawing 33 . If the record count per truck for every estimated case is calculated, it will become like drawing 13 - drawing 16 .
[0062] Moreover, the memory space estimate for every gap deletion approach is as follows.
< case All gaps also including I-1> and an inter-record gap will be deleted.
- Format 1 (case of the most record counts)
size of HA Size of =40-12=28R0 Size of =R0C+R0D=(40-12)+1=29Rn =RnC+RnD=(40-12)+1=29 -- therefore -- the capacity for one truck -- HA+R0+Rnx93=28+29+29x93=2754- format 2 (type of a journal file)
size of HA Size of =40-12=28R0 Size of =R0C+R0D=(40-12)+8=36Rn =RnC+RnD=(40-12)+200=228 -- therefore -- the capacity for one truck -- HA+R0+Rnx68=28+36+228x68=15568- format 3 (paging, swapping, VSAM)
size of HA Size of =40-12=28R0 Size of =R0C+R0D=(40-12)+8=36Rn =RnC+RnD=(40-12)+4096=4124 -- therefore -- the capacity for one truck -- HA+R0+Rnx10=28+36+4124x10=41304- format 4 (case of the minimum record count)
Size of HA Size of =40-12=28R0 Depending =R0 C+R0D=(40-12)+47988=48016, the capacity for one truck is HA+R0=28+48016=48044[0063]. < case It leaves only I-2> and an inter-record gap, and the gap between the fields is removed (ECC, padding, etc. remove). However, if gap length will be lengthened for the part and it will be made consistent even if it deletes the gap between the fields, ECC, a tooth space (X'FF'), padding, etc., memory space will become completely the same as the original CKD format. Therefore, deleted consistency doubling estimates in the case which is not performed here.
- Format 1 (case of the most record counts)
size of HA Size of =G1+HA=504+(40-12) =532R0 Size of =G2C+R0C+R0D=248+(40-12)+1=277Rn =G3+RnC+RnD=216+(40-12)+1=245 -- the capacity for one truck therefore HA+R0+Rnx93=532+277+245x93=23594- format 2 (type of a journal file)
Size of HA Size of =G1+HA=504+(40-12) =532R0 Size of =G2 C+R0 C+R0D=248+(40-12)+8=284Rn It depends =G3+RnC+RnD=216+(40-12)+200=444. The capacity for one truck is the HA+R0+Rnx68=532+284+444x68=31008- format 3 (paging, swapping, VSAM).
Size of HA Size of =G1+HA=504+(40-12) =532R0 Size of =G2 C+R0 C+R0D=248+(40-12)+8=284Rn It depends =G3+RnC+RnD=216+(40-12)+4096=4340. The capacity for one truck is the HA+R0+Rnx10=532+284+4340x10=44216- format 4 (case of the minimum record count).
Size of HA Size of =G1+HA=504+(40-12) =532R0 Depending =G2 C+R0 C+R0D=248+(40-12) +47988=48264, the capacity for one truck is HA+R0=532+48264=48796[0064]. < case It leaves I-3> and not only an inter-record gap but the gap between all the fields (ECC, padding, etc. remove). If gap length will be lengthened for that part and it will be made consistent even if it deletes ECC, a tooth space (X'FF'),

padding, etc. also in this case, memory space will become completely the same as the original CKD format. Therefore, deleted consistency doubling estimates in the case which is not performed here.
- Format 1 (case of the most record counts)
size of HA = Size of G1+HA=504+(40-12) =532R0 Size of =G2 C+R0 C+G2+R0D=248+(40-12) +224+1=501Rn = It depends G3+RnC+G2+RnD=216+(40-12)+224+1=469. The capacity for one truck is the HA+R0+Rnx93=532+501+469x93=44650- format 2 (type of a journal file).
size of HA = Size of G1+HA=504+(40-12) =532R0 Size of =G2 C+R0 C+G2+R0D=248+(40-12) +224+8=508Rn = It depends G3+RnC+G2+RnD=216+(40-12)+224+200=668. The capacity for one truck is the HA+R0+Rnx68=532+508+668x68=46464- format 3 (paging, swapping, VSAM).
size of HA = Size of G1+HA=504+(40-12) =532R0 Size of =G2 C+R0 C+G2+R0D=248+(40-12) +224+8=508Rn = It depends G3+RnC+G2+RnD=216+(40-12)+224+4096=4564. The capacity for one truck is the HA+R0+Rnx10=532+508+4564x10=46680- format 4 (case of the minimum record count).
size of HA Size of =G1+HA=504+(40-12) =532R0 =G2 C+R0 C+G2+R0D=248+(40-12) +224+47988=48488 -- therefore -- the capacity for one truck -- HA+R0=532+ -- 48488= 49020 [0065] It is < case to drawing 17 . I-1> - < case About three kinds of gap deletion approaches to I-3>, the memory space when developing four track format to formats 1-4 on memory, respectively is shown. Drawing 18 expresses this with a graph.
[0066] Every track format is <a case I-1> - < case so that drawing 17 and drawing 18 may show. Memory space is also mostly needed, so that there are many I-3> and gaps to leave. < case In I-3>, in order to leave all gaps, in every format, memory space almost equal to the capacity in the original CKD format is needed. < case where the gap between the fields is deleted When I-2> also takes the method with which only the part of the gap between the fields from which it deleted as a check method of track capacity, ECC, and padding lengthens an inter-record gap, and maintains the relative position from the head of the truck of each record with the original CKD truck, required memory space is < case. It becomes almost the same as I-3>. On the other hand, < case where all gaps are deleted at I-1>, memory space is small notably in the formats 1 and 2 especially with small record size. In the format 1 of 93 records, 16 - 1/17 of other methods and format 2 are also about 1/3. Thus, the following merits can be considered that there is little memory space for developing a truck image, and it ends (refer to drawing 19).
(1) If the unit of management of cache memory is made into the finer unit (for example, the unit which writes to one data disk = a part for 1/4 truck ... 12KB) instead of a part (about 48KB) for one truck of CKD, the data of other trucks can be written to the surplus field of cache memory. therefore, the data of the truck of many in the same cache memory space -- it can hold -- the improvement in a utilization factor of cache memory -- lengthening -- **** -- improvement in a hit ratio is expectable.
(2) Also when writing the data to a physical disk, I hear that that there is little amount of data of the image for one truck, and it ends ends by the small amount of data, and there is. For example, if the data of the image for one truck end by one half of CKD formats, the data transfer to a physical disk will require data disks 1 and 2 and only a parity disk, and if it is not a format light, it is not necessary to transmit data to data disks 3 and 4. This is connected with unloading, such as each bus in a subsystem. However, although memory space ends few, control will become very complicated, if the field on a physical disk doubles and is not secured to maximum. After writing the data of other trucks to the field which was vacant also when writing the data to a disk just because the truck image on memory is the one half of a CKD format, when the truck is reformatted next and the amount of a truck image has increased, it becomes what carries out a light in search of the field which it stops could carry out the light of the truck image to the field to which it continued on the disk, and is vacant on the disk somewhere. Although it is more good if the free area of the

part physical disk is also effectively utilizable when the amount of data of a truck image becomes less, at least the merit of (1) and (2) is [but] attractive enough.
As mentioned above, as a result of actually estimating the size of memory space, in the case where all gaps are left, and the case where all gaps are deleted, it turns out that the aperture of several times or more comes out [required memory space] depending on record count.
[0067] Next, "the memory space estimate of the positional information of a record" is explained. As a method which positions the purpose record, it is < case. II-1>, and <case II-2> and < case Three methods of II-3> were held. It is < case among these. Each of other two methods except II-1> is methods holding the control information which indicates the location of the record on memory to be the record itself separately. Here, it estimates at the control information for this positioning how much memory space (capacity on =FBA disk) is the need by the all directions formula. Here, it is < case. < case except II-1> II-2> and < case It estimates about II-3>. moreover, < case In II-3>, although there are an approach (for example, head of a truck) of the positional information of a record bundling up a part for all records, and holding to one place and the approach (for example, every -- the information which shows the location of the record contained in the FBA block at the head of a FBA block is held) of holding dispersedly, it estimates about the approach of holding dispersedly here. The reason which is not estimated about the approach of bundling up a part for all records and holding to one place is as follows. Unless it finishes reading the data from the disk with which positional information is recorded even if it finishes reading the data from the disk with which the purpose record is recorded when an array mold disk unit like drawing 19 is considered and the data for 1CKD truck will be read from four sets of data disks, if positional information is collectively held by one place, the record location of the purpose record cannot be found and data transfer with a host cannot be started. Therefore, when a rotation synchronization is not able to be supported, or when [even if it is supporting,] a standby disk is added and the synchronization has collapsed, it leads to the increment in average rotational delay.
[0068] Here, let estimated conditions be the following. First, 4 bytes of information per record as shown in the following figure shall be held as maintenance information per record.
Record No.: It is record No. of the CKD record currently recorded. < case Although it is not necessarily indispensable on control in II-2>, it is especially < case. At II-3>, if it has this record No. in control information, even if it does not read the count field of an actual record one by one, the purpose record can be discovered by one shot.
Sector value : The sector value in the case of memorizing as a CKD record is memorized. The comparison with the sector value specified by this sector value by the set sector command is attained. Moreover, it can ask for the relative position from the truck head of that record with this sector value, and the already consumed capacity can be judged. Since the sector consists of seven segments, it can calculate a segment number using this sector value. That is, it is segment number = sector value x7.
Memory address: It is a byte address on the memory of the CKD record currently recorded. Since addressing can be carried out to 64KB if it 2B Is, a format of drawing 33 is enough. It does not ask whether the origin of the address is made into the head of a CKD truck, or it is made heads which are the management units of positional information, such as the every FBAblock, for the time being.
[0069] Moreover, as a unit (unit holding the positional information of a record) which manages the positional information of a CKD record, it estimates about two cases of drawing 21 . It sets in the two above-mentioned cases, and is < case. II-2> and < case The memory space for having the positional information of the CKD record currently recorded about each of II-3> is estimated. The estimated result is as follows.

< case II-2> : Only the positional information of a top record is held. In this method, since only the positional information for every one management unit record is always held irrespective of the size of the management unit of positional information, the memory space for positional information is 4B for every management unit.
< case II-3> : The positional information of all records is held. By this method, the record count of CKD which may be recorded there changes with magnitude of a management unit. Furthermore, what CKD record is recordable per management of a certain size is < case where record count increases most here although it is dependent also on the deletion approach of a gap. It estimates by adopting I-1> (all gaps being deleted). The size of Rn of the "format 1 (case of the most records) of < case I-1>" of "the deletion approach of a gap and evaluation of memory space" which the minimum memory space per one record at this time mentioned above shows that it is 29B. Therefore, the most record counts NR recordable for every size of each management unit are if memory space for positional information is set to alpha. NR= {(size of a management unit) -alpha} /29B ... (1)
Moreover, since what is necessary is just to hang 4B on the record count NR, memory space alpha of the positional information per management unit is alpha=NRx4B. ... (2)
It comes out and asks. When these two formulas are solved, it is NR= (size of management unit)/33 (below decimal point omission)... (3)
It becomes. Therefore, alpha is also immediately called for by substituting for (2) types NR calculated by (3) formulas. However, since the record count in 1 truck does not exceed 94 pieces in the track format of drawing 33 (R0 is included), the maximum of NR is also 94. If the above result is summarized, it will become like drawing 22 .
[0070] in order to record a CKD record according to "the deletion approach of a gap and evaluation of memory space" which were mentioned above -- net -- the data field of the maximum of required memory space is the case where it is the longest, by 1 truck 1 record (only R0). this time -- required memory space -- HA from drawing 17 -- it doubles R0 and becomes like drawing 23 . If capacity of the FBA disk prepared for one CKD truck is set to 48KB (12KBx4), the memory space which can be used for control information will become like a bottom type.
in < case II-2>, it is said that memory space runs short irrespective of the size of a management unit -- it is satisfactory. I-1> 1024x48-48044=1108 < case I-2> 1024x48-48796= 356 < case I-3> 1024x48-49020= 132 -- if this value is compared with the value (column of most right-hand side) of drawing 22 -- < case On the other hand, < case < case where a management unit is large (management unit 2 size 12KB), and there is least memory space the case (1504B) where there is little memory space for positional information, and for a record, and it can be managed with II-3> Even if compared by I-1>, memory space about 400 B which can be used for positional information will run short.
[0071] < case Since there is little II-3> and its count of memory access for looking for the purpose record is more promising than the conventional method (< case II-2>), it is necessary to consider a cure to this insufficiency. The following can be considered as this cure.
Cure 1 The capacity reserved in per physical disk is increased.
ex. If 12KB -> 13KB, 52KB, a next door, and lack will be solved for the capacity which can be used for one CKD truck at once. .... Although a physical disk is [52 KB/CKD truck] necessary and disk area becomes useless compared with 48 KB/CKD truck, since the cost of a physical disk has been decreasing, it is one of the effective approaches.
Cure 2 The amount of control information required for per record is reduced.
ex. 4B / record -> when it is 2B/record, the total memory space required for control information is set to

752B, and stops running short.
Cure 3 The number of control information is reduced.
ex. In the case of the management unit 2 (size 12KB), by this estimate, I think that the positional information of a maximum of 94 records is held every four management units. However, since the Max.94 piece is required of the sum total of four management units in practice, it considers devising well and reducing the total capacity of control information.
[0072] The result of "the memory space estimate of the positional information of a record" is as follows as mentioned above. < case In II-2>, memory space runs short and is satisfactory. < case Although it changes also with conditions in II-3>, the memory space which can be used for positional information will run short a little. < case Since there is little II-3> and its count of memory access for looking for the purpose record is promising, it is necessary to consider a cure to this insufficiency from now on.
[0073] Next, "an improvement of the control system of the positional information of a record" is explained. The trouble of the control system of a previous estimate is as follows. In the "memory space estimate of the positional information of a record" mentioned above, the field of positional information was secured to every [which manages the positional information of a CKD record] unit (unit holding the positional information of a record) according to the maximum record number which may be recorded there. As shown in drawing 24 , when setting size of a management unit to 12KB, speaking concretely, a maximum of 94 CKD records may be recorded there. Since 4 bytes was estimated as positional information per record, the positional information field of only 4Bx94=376B had been estimated for 12KB of every management unit. The truck image of CKD truck 1 duty consists of 12KB of these four management units, and since data are distributed and recorded on four sets of these 12KB every data disks, the memory space of the positional information per CKD truck will be set to 376Bx4=1504B. However, in this, the field of the positional information for 4= 94x376 record will be secured per CKD truck. In a format of drawing 33 , since the maximum record numbers of 1CKD truck are 94 records (it contains R0), even if the maximum record number which may be recorded on 12KB of each management unit is 94 pieces, respectively, every 94 records must have been recorded on coincidence by not no management units. Therefore, the field of the positional information for 3= 94x282 record will be useless.
[0074] Such a trouble is generated from it having made immobilization size of the positional information field within the management unit of a record. Since the field of memory where a CKD record can be written in also at the time of that control is easy and (2) format light since the size of the positional information field over which it must jump is fixed also when performing data transfer of the record currently recorded ranging over (1) management unit whenever it makes size of a positional information field immobilization is immobilization, advantages, like control is easy occur. As long as size of a positional information field is made immobilization, the size does not obtain a doubling colander to maximum, but has no choice but to secure [every management unit] for 94 records. On the other hand, although control only whose part of the record actually recorded as making size of a positional information field adjustable uses a positional information field is also possible, the above advantages will be lost fundamentally. Although it is indispensable to make adjustable size of the positional information field of each management unit in order to reduce the total capacity of positional information, it is important to avoid such a fault well.
[0075] The improvement proposal which made size of a positional information field adjustable at drawing 25 is shown. Hereafter, an improvement proposal is explained based on this drawing. In addition, the size of the management unit of the positional information of a record is dealt with henceforth about the case of the same 12KB as the size of the division unit to a data disk. In the same 1-2KB as the block (sector) size of the same FBA disk as the conventional method, it is because effectiveness is bad (for details, refer to "the

problem of a unit with the positional information of a record" mentioned above). Moreover, henceforth, as shown in drawing 25 , the thing of the positional information of a control frame and a record calls a record pointer the field where, as for the thing of the management unit of the positional information of a record, the thing of a frame and the field which actually records a CKD record records a record frame and control information. The points of this method are the following points.
(1) Only the part of the record actually recorded uses a record pointer. Thereby, the size of a record pointer field can be managed per 1CKD truck and with 4bytex94=376byte at the maximum.
(2) Hold the size of the control frame containing a record pointer as control information (control frame pointer of drawing 25) separately. Using this information, even if the size of a control frame serves as adjustable, it is not necessary to complicate control.
(3) Record such control information (control frame) on the tail end instead of a head of each frame. Moreover, the record pointer in a control frame is also arranged in an order from back. (The record pointer corresponding to it in what has an actual record in front within a record frame comes back within a control frame) . Thereby, control at the time of a format light can also be performed easily.
[0076] Hereafter, according to a motion of actual read/write, the approach of the point-to-point control of the record in a frame is explained to a detail to a slight degree.
(1) If a frame is read from the (a) physical disk on cache memory at the time of lead/update light, CPS reads the control frame at the tail end. Although it does not understand at this time until it reads the size of a control frame, the suitable byte count at the tail end of a frame is read collectively for the time being. (The byte count read at this time needs to choose an optimum value by architecture, such as CPS, and cache memory, a bus.) For example, when tail end 32B of a frame will be read and the record count currently recorded in this frame is seven or less pieces, CPS can acquire the record pointer of all records by this one access. For example, when making 4KB size into most record sizes, since the record count in one frame is three or less pieces, in most cases, all record pointers can be acquired by this one cache memory access.
(b) From the memory address in the control frame pointer at the tail end of the read control frame, the control frame was still read, and it has left it, or can judge in no. If it has read and left the control frame, CPS will read it. By this, even when CPS is the worst, all record pointers can be acquired by two memory accesses.
(c) Next, find the memory address on the frame of the purpose record from the argument of the search sent by the host, and the acquired record pointer (it means that orientation was established now).
(d) Compare with the memory address of the purpose record the size of the count field, the value (memory address at the tail end of the purpose record) which added KL and DL, and the memory address which shows the head of the control frame in a control frame pointer before starting data transfer among hosts. When the memory address at the tail end of the purpose record is eating into the control frame, after CPS carries out an end halt of the data transfer with a host before a control frame and advances a memory address to the head of the following frame, it performs control which resumes data transfer with a host. If the tail end of the purpose record is settled in the record frame, the above control is unnecessary, and CPS can be completed, without interrupting the data transfer of the purpose record on the way.
(e) Since the following record is continuing on memory as it is succeedingly when the command of lead/update light system is taken out from a host, even if it does not use a record pointer, data transfer can be continued as it is. However, the comparison of the memory address at the tail end of a record and the start address of a control frame must be performed for every record.
[0077] (2) If a frame is read from the (a) physical disk on cache memory when adding a record within the

same frame at the time (2.1) of a format light, CPS reads the control frame at the tail end.
(b) Investigate a control frame pointer and acquire all record pointers.
(c) Next, find the memory address on the frame of the purpose record from the argument of the search sent by the host, and the acquired record pointer (it means that orientation was established now).
The above (a) - (c) procedure is the same as the time of lead/update light.
(d) Write sent by the host It investigates whether track overrun is generated from the value of the count field of CKD. (This detailed procedure is separately explained as "a check method of track capacity".) When generating track overrun, even the track overrun point performs data transfer by subsequent control, and control which reports track overrun to a host is performed. If the check of track overrun finishes, the memory address which shows the head of the control frame in the control frame pointer which CPS holds by itself will be 4B Reduced (4B escape of a control frame is done), and the record pointer of the record added still more newly will be added to the head of the control frame which CPS holds.
(e) Write The memory address at the tail end of the record newly added from the value of the count field of CKD is calculated, and it compares with the memory address in the control frame pointer updated by (d). When the memory address at the tail end of a new record is eating into the control frame, after CPS carries out an end halt of the data transfer with a host before a control frame and advances a memory address to the head of the following frame, it performs control which resumes data transfer with a host. If the tail end of a new record is settled in the record frame, the above control is unnecessary, and CPS can be completed, without interrupting the data transfer of a new record on the way.
(f) Since the following record is continuing on memory as it is succeedingly when the WriteCKD command is taken out from a host, even if it does not use a record pointer, data transfer can be continued as it is. However, the comparison of the memory address at the check of track overrun, renewal of a control frame pointer, addition of a record pointer, and the tail end of a record and the start address of a control frame must be performed for every record.
(g) When a command is completed, write out the control frame after updating which CPS held on cache memory.
[0078] In addition, in a case as shown in drawing 26 , when it is going to add a record and a record pointer is added into the succeedingly same frame just because 3 bytes of record frame is vacant, a record pointer will eat into the posterior part of a front record. That is, when the availability of a frame is 4 bytes or less, into the same frame, a new record cannot be added but must use the following frame. Thus, it is vacant and having included into the control frame and having managed tends to give [which has been produced / a jump of the control frame at the time of the data transfer of (e) etc.] a cutting tool (4 bytes or less). Therefore, the memory address in a control frame pointer will show this empty cutting tool's start address. The offset (0-4 bytes) from this empty cutting tool's head to the start address of a record pointer is held as control information in a control frame, in the case of acquisition of the record pointer of (b), lengthens this offset value from the memory address in a control frame pointer, and should just ask for the start address of a record pointer.
[0079] (2.2) When writing a record newly [when writing a record first within the same frame] in a format light on the case where it reformats completely newly from the head of a truck, and the frame, on which one was not recorded for the record, CPS will generate a control frame newly. Therefore, the establishment activity of the orientation by acquisition of the control frame of (a) - (c) of (2.1) is unnecessary in this case. The check of the track overrun of (d) is performed like (2.1).
(e) If the check of track overrun finishes, CPS will generate a control frame pointer (the start address of a control frame shows the place for one record pointer.) on its memory, and the record pointer of the record

written still more newly will be written to the head of the control frame which CPS holds.
(f) Write The memory address at the tail end of the record newly written from the value of the count field of CKD is calculated, and it compares with the memory address in the control frame pointer generated by (e). When the memory address at the tail end of a new record is eating into the control frame, after CPS carries out an end halt of the data transfer with a host before a control frame and advances a memory address to the head of the following frame, it performs control which resumes data transfer with a host. If the tail end of a new record is settled in the record frame, the above control is unnecessary, and CPS can be completed, without interrupting the data transfer of a new record on the way.
(g) It is Write from a host succeedingly. When the CKD command is taken out, it becomes being the same as that of (2.1).
(h) When a command is completed, write out the new control frame which CPS held on cache memory. In processing of a format light system, unless it is after the record count recorded on the frame is known, the size of a control frame is not decided. Therefore, it will be understood that it should be begun [in / for a control frame / the head of a frame] from where [of a frame] to write the first record.
By this method, the control frame has avoided such a problem by beginning to write from the tail end of a frame by beginning to write a record from the head of a frame as mentioned above.
[0080] Next, memory space of positional information with this improvement proposal is estimated. It estimates about the case of the management unit 2 of drawing 22 among the cases estimated by the "memory space estimate of the positional information of a record" mentioned above about the memory space of positional information with this improvement proposal, and an improvement effect is seen. The management unit (frame) size of this case is 12KB, and record count has estimated memory space about the case where they are 94 records / CKD truck with which positional information (record pointer) increases most. In this case, although based also on the deletion approach of a gap, in the case where all gaps are deleted, all records are settled in a one-eyed frame (refer to "the deletion approach of a gap, and evaluation of memory space"). at this time, with this improvement proposal, 94 pieces and one control frame pointer have a record pointer in the control frame of a one-eyed frame, and there is especially no control information in other frames (rather than -- other frame itself is not needed). Therefore, memory space required for control of positional information is 4Bx95=380B. The comparison with this result and other cases estimated by "the memory space estimate of the positional information of a record" is shown in drawing 27 . < case II-2> is the conventional method and is a method only holding the location of the record located in the head of each FBA block (here frame). < case an improvement II-3> is a method holding the positional information of all records, and according to a book -- the front -- it is a formula. < case II-3> ** is this method.
[0081] as "the memory space estimate of the positional information of a record" was shown, in order to record a CKD record -- net -- the data field of the maximum of required memory space is the case where it is the longest, by 1 truck 1 record (only R0). At this time, if required memory space is quoted from drawing 23 , it will become like drawing 28 . If capacity of the FBA disk prepared for one CKD truck is set to 48KB (12KBx4 set), the memory space which can be used for control information will become like a bottom type. < case [ ] -- I-3> 1024x48-49020= 132 -- except < case I-1> which all deletes a gap as compared with this value and memory space 380B of the positional information of < case II-3> ** of drawing 27 , it seems that memory space becomes less insufficient. I-1> 1024x48-48044=1108 < case I-2> 1024x48-48796= 356 < case However, the upper value is 1 truck 1 record to the value of drawing 27 being a value in case record count of memory space of positional information (control frame) increases most by 94 records, and it is a value in case the memory space for a CKD record is max. With this improvement proposal, if record count

decreases, the memory space of a control frame will also decrease proportionally and it will end with the case of 1 truck 1 record by 20B (one record pointer, four control frame pointers). After all, this improvement method will be sufficient for memory space about all cases. ((however, less insufficient) If the method which lengthens a gap and adjusts a part to have deleted [padding / ECC,] is taken as a check method of track capacity in order to hold the relative position of < case III-3>, i.e., a CKD format of each record, memory space, such as a case of 94 records, may become) The memory access for positioning of the purpose record can also be managed with one - 2 times per frame again. Once CPS acquires a control frame from cache memory at this time, since CPS should just use this acquired control frame in next control, the memory access for control is unnecessary. What is necessary is to update the control frame which CPS acquires and just to write out to cache memory finally, also when the configuration of the record in a frame changes in a format light etc. That what is necessary is just to check the memory address in the control frame pointer with which CPS holds jump control of the control frame under data transfer with a host, the control at the time of a format light, etc., since such actuation is performed in the time amount of an inter-record gap, it does not become an excess head. As mentioned above, this improvement method can say that the early purpose attained enough.

[0082] Next, "the check method of track capacity using a segment number is examined." As the three points by which a CKD-FBA conversion method is characterized, although the check approach of the positioning approach <III> track capacity of the deletion approach <II> record of the <I> gap was raised, especially here, it is < case about the check approach of the track capacity of <III>. The actual control approach is explained about the method which checks track capacity using the segment number of III-3>. In addition, about the check approach of the track capacity of <III>, it is < case. III-1> and < case III-2> and < case Three methods of III-3> were going up. First, a prerequisite is considered as follows. A segment number uses 32B as one segment like drawing 35 . A segment number shall use the absolute segment number (ASN) which it **ed in an order from 1 (refer to drawing 29). Moreover, all the figures to be used use a decimal number.

[0083] Next, the control approach is explained. The following two points are first considered as a statement of principles.

(1) Since the segment number of the home address HA can be written from S/W, two kinds, the case (ASN=17) where HA is in the usual location, and when HA is Mov(ing) (ASN=23), shall be supported.

(2) Since neither a segment number nor SC (skip control) can be written from S/W about other records containing the other R0, there shall be neither Move nor Split.

(1) Based on how to give a segment number, next the above-mentioned statement of principles, the segment number of each record shall be given as follows.

(a) what converted absolutely into the segment number the value in which the segment number of HA is written from S/W -- following -- ASN= -- it is referred to as 17 or 23 (of course, it is made into ASN=17 at the time of initialization.).

(b) The segment number of R0 is always set to ASN=26 irrespective of the segment number of HA.

(c) Define the segment number of the record after R1 (Rn) by the bottom type.

- When key field are in a front record (Rn-1) and there are no key field in the record (Rn-1) of ASN+ [(KL+12) /32]+[(DL+12) /32]+22, and [of ASN=Rn -1 of Rn] a front, It corrects ASN+[(DL+12) /32]+of ASN=Rn -1 of Rn 15. The byte count of ECC and a tooth space and 32 12 The byte count of a segment, the number of segments of 22G2 (224B), G2 (224B), G3 (216B), and the count field (40B), and the number of segments of 15G2 (224B), G3 (216B), and the count field (40B) -- it is -- [] -- inside is considered as the below decimal point up valuation.

[0084] (2) Check track overrun by the bottom type using the segment number for which it asked by the check approach (1) of track capacity (track overrun), and KL and DL which are sent by the host. If the bottom type is filled, it OKs, and it will become track overrun if it is not filling.
- When key field are in the record (Rn) to add and there are no key field in ASN+[(KL+12) /32]+ [(DL+12) /32]+14 <=1533 and the record (Rn) to add of Rn, ASN+ [(DL+12) /32] of Rn -- +7<=1533 however the number of segments of 14G2 (224B), the number of segments of 7G2 (224B), and 1533 -- the number of the maximum segments -- it is -- [] -- inside is considered as the below decimal point up valuation.
[0085] When the above is summarized, the procedure of the check of the track overrun at the time of a format light is as follows. It asks for the segment number of the record to add according to the procedure of (1).
- Confirm whether, according to the procedure of (2), the record to add starts track overrun from KL.DL sent by the value and host of the segment number for which it asked. The same formula is used for it, although HA will Move each type of the above (1) and (2) and it will not be. That is, although HA will Move on an imagination CKD truck and it will not be by this method as shown in drawing 30 , all defects will be in the tail end of a truck irrespective of the value of SC in HA (skip control). Since the need of caring about the value of SC (skip control) in being able to use the always same formula with the check of track overrun by making it such control is lost, control becomes very simple. Whenever it has shifted R0 in accordance with Move of HA, at the time of the check of track overrun, the segment number of HA of the head of a truck is read and checked, or the need of holding and managing SC (skip control) also in each record comes out like an actual disk. Fundamentally, since the activity in the gap at the time of a format light is the direction which increase in number compared with the conventional disk control, such as a check of the track overrun of (2), the part made simply should be made simple.
[0086] As mentioned above, the approach of the check <case III-3> of the track capacity which used the segment number is realizable by comparatively simple control. Therefore, conventional method < case Even after changing the truck image of a CKD format into FBA like the approach of III-2>, it is controllable, even if it adjusts gap length and does not maintain the byte position from the head of the truck of each record by force with the original CKD truck. The segment number itself is < case where it is positively used since the support on a S/W interface is required (if attached to HA at least). The approach of III-3> < case where only an amount equivalent to the original CKD format must hold gap length compared with III-2>, it is remarkably advantageous in the use effectiveness of cache memory when especially record size is small ("the deletion approach of a gap, and evaluation of memory space"). Moreover, < case which calculates the remaining capacity of a truck by reading KL.DL of all records To III-1>, it is advantageous in respect of the count of access of memory, and is the most promising method.
[0087] As the three points by which a CKD-FBA conversion method is characterized, the check approach of the positioning approach <III> track capacity of the deletion approach <II> record of the <I> gap was raised, and the following examination has so far been performed about each.
- In - "the deletion approach of a gap, and evaluation of memory space", memory space was evaluated about the deletion approach of the gap of <I>. Consequently, < case where all gaps are deleted It turned out that the method of I-1> is advantageous.
- Memory space for positional information was estimated about two methods which hold the positional information of a record by - "the memory space estimate of the positional information of a record" about the positioning approach of the record of <II>. Consequently, it is < case at the point of the count of memory access. < case made more advantageous than II-2> (only the positional information of the record of the

beginning within a FBA block is held) If II-3> (the positional information of all records is held) remained as it was, it turned out that the memory space for holding the positional information of a record increases too much.
- It is < case by "improvement of the control system of the positional information of a record" there. It examined whether the improvement of the memory space of II-3> would be possible. Consequently, it is < case by devising the control approach. It turned out that memory space for II-3> to also hold the positional information of all records can be made sufficiently small.
- It is -< case about the check approach of the track capacity of <III>. III-3> (the relative-position information on a CKD format of each record is held using a segment number etc.) a part with an unnecessary gap -- "examination of the check method of track capacity using a segment number" examined feasibility [being advantageous]. Consequently, it was easy to check track capacity using a segment number, and it also showed the concrete procedure.
[0088] As shown in above drawing 19 which carried out the examination result above-mentioned, it turned out that the method <1, 3, 3> which is getting the high score most is employable. Especially, since this method deletes a total gap as the deletion approach of a gap, when record sizes, such as a journal file, are small, there is little memory space for storing a truck image, and it ends. Since the method holding the positional information of all records is adopted as a positioning method of a record, there are few counts of memory access for looking for the purpose record, and they end. However, < case examined by "improvement of the control system of the positional information of a record" in order to lessen memory space for holding positional information It is necessary to adopt II-3> **. Since it is the method which uses a segment number as a check method of track capacity, there is no need, therefore concomitant use with the approach of deleting a total gap of the gap for maintaining with a CKD format of the relative position of each record is attained as the deletion approach of a gap. It does not judge only by the size of a score but the comparison with a method <1, 3, 3> is performed about other methods which are getting the high score.
[0089] <2, 3, 2> ... In 54-point drawing 9 , as compared with <1, 3, 3>, a segment number was not used for the method <2, 3, 2> which is the runner-up by the check approach of track capacity, but it has taken the method which adjusts the gap length who is the conventional method and maintains the relative position in a CKD format of each record also on memory. Therefore, a total gap cannot be deleted but the deletion approach of a gap serves as a method which leaves some gaps inevitably. Since the good point of this method is maintained while the relative position of each record has been a CKD format, it is not checking track capacity according to it in calculating a segment number **** at the time of a format light. Therefore, the segment number itself is omissible. in order for a bad point to lengthen gap length and to maintain the relative position in a CKD format of each record -- the track capacity on memory -- always -- almost -- a CKD truck -- it is becoming required 1 duty. Especially, this leads to use effectiveness aggravation of cache memory, and load increase of the internal bus at the time of a staging/destaging, when record sizes, such as a journal file, are small. Moreover, although the method holding the positional information of all records is used together as a positioning method of a record, a record size may serve as an insufficient memory space very much by the minimum in the improvement approach examined by "examination of the check method of track capacity using a segment number" in the case of 94 records / truck of the most numerous [record count]. moreover -- although the positioning method of a record differs from the conventional method in this method -- the check method of track capacity -- the conventional method -- it remains as it is.
[0090] <2, 3, 3> ... the 53-point this gentleman type is the same as <1, 3, 3> except leaving a part of gap.

Therefore, since it is not used especially even if it leaves a gap, it is the method which does not have the semantics which leaves a gap and does not have the necessity adopted specially compared with <1, 3, 3>. By the above-mentioned explanation, although premised on the array mold disk unit (RAID), also when using the usual disk unit, it states that the method of <1, 3, 3> is employable here.
- That the size of the memory space by the deletion approach of a gap is greatly concerned with the use effectiveness of a cache about the deletion approach of the gap of <I> is < case where it is the same even if it is RAID and there is nothing, and all gaps are deleted from this point. There is no change in the method of I-1> being advantageous.
- It is better to perform data transfer of the amount which is one bus acquisition and which was collected the grade, in order to make a high transfer rate easy to lower the load of a bus about the positioning approach of the record of <II>, and to take out. That is, the positional information of a record also has the direction more advantageous than the method which accesses the count field of each record each time which can be accessed collectively. Therefore, < case with few [approach / of a record / positioning] counts of memory access The direction of II-3> (the positional information of all records is held) is < case of the conventional method. It is more advantageous than II-2> (only the positional information of the record of the beginning within a FBA block is held).
- About the check approach of the track capacity of <III>, even if this is also no longer RAID, there is no change in the method with an unnecessary gap being more advantageous on the use effectiveness of a cache. Therefore, < case III-3> of the conventional method From III-2> (gap length is adjusted and the relative position in a CKD format of each record is maintained also on memory), it is < case. (the relative-position information on a CKD format of each record is held using a segment number etc.) a part with an unnecessary gap -- advantageous -- it is -- it is good as.
[0091] Next, the command emulation method in CKD-FBA conversion is examined. Here, based on the concrete conversion method mentioned above, the implementation approach in this CKD-FBA conversion method is examined for every command typical [on an actual S/W interface].
[0092] 1. SET 1 byte of sector information is retrieved for reception and a sector from a SECTOR <function> channel.
(Sector information: X'00' - X'DD, X'FF')
<Implementation method> In this disk subsystem, a sector value is used in order to know in which frame the purpose record exists first among the frames which divided the CKD truck into four. After a frame is found, high positioning of precision can be performed more by comparing the segment number and this sector value of each record currently written to the record pointer in a control frame (although the outline of a frame and a control frame, or a record pointer was shown in drawing 25 , improve " refer for the control system of the positional information of "record to for details).
O count of the frame in which the purpose record exists -- by the actual CKD disk, since the amount of one sector consists of 224 bytes, the absolute byte address which is equivalent to it on the truck of a CKD format from a sector value is calculable immediately. However, since a gap, ECC, padding, etc. are deleting and recording on a frame, unless it converts the part to have deleted, it is not known in which frame the absolute byte address is included. It depends for this eliminated daily dose also on record count.
[0093] Hereafter, count of the frame in which the purpose record exists is explained. On a frame, it does not become settled what byte is deleted from the truck of a CKD format, if it is not decided that the track format on a frame will be a detail, but supposing it is a format like drawing 33 , the byte count deleted on the frame for every field will become like drawing 31 . Since the record of an individual (record number 1 of the purpose record) exists except for R0 in front of the purpose record, the location on the truck of the CKD

format shown with a sector value is equivalent to the location expressed with byte-count BC1 of a bottom type from the head of a truck on a frame.
BC1=224x sector value-1038-752x (record number 1)
Moreover, in order to record the record of only this, on a frame, the record pointer for record count (R0 is put in and it is a record number individual) is required. Since one record pointer is made into 4 bytes, byte-count BC2 included also at this rate becomes like a bottom type.
If 4 bytes of control frame pointer is removed on one BC2=BC1+4x record number frame, since it can do 12KB-4B record also including a record pointer, it is expectable that the location equivalent to a set sector value exists in the Nth frame called for by the bottom formula.
N=[BC2/12284]
however, [] -- inside is considered as the below decimal point up valuation. If these formulas are summarized to one, it will become like a bottom type.
N= [(224x sector value-286-748x record number) /12284]
however, [] -- inside is considered as the below decimal point up valuation. Moreover, it is referred to as N= 1 when it becomes a negative value. Since the byte counts deleted on the frame therefore differ correctly in the byte count which is carrying out zero padding at KL and DL in order to make [whether key field exist in each record, and] it the integral multiple of 32 again, an upper type turns into a rough type to the last. By the upper formula, since it has assumed and calculated about the part of such an indeterminate so that the byte count deleted may become max, the number of the frame for which this asked actually becomes the inclination which shows the front also for a twist. Therefore, what is necessary is just to look for the following frame, when the purpose record does not exist in the frame for which it asked by the upper formula. However, the precision of this formula is good from the first at size 12KB of a frame, and since an amount with error is 286 bytes (224+31+31) at the maximum per record to it, it does not usually have a problem.
[0094] Moreover, it is SEARCH, even if the SEARCH command does not come or it is the SEARCH command. Since it was KEY, when a record number is not obtained, it will give up specifying a frame and the segment number of the record in the frame (for example, frame which the staging completed first) of arbitration will be investigated (refer to the procedure of the following term).
O With the procedure on retrieval of the purpose record, when the frame in which the purpose record exists is specified, the rest investigates the record pointer in the frame, and should just look for the purpose record (a detailed procedure is "improvement of control system of positional information of record" reference). First, the sector value currently recorded into the record pointer in an order from the record pointer (it is in the tail end in a control frame) of the record of the head of the frame is investigated. And the record pointer of the first record with a larger sector value than the sector value sent from the channel is found. If all the records in this frame have only the sector value smaller than the sector value sent from the channel, the same actuation is performed to the following frame. If the record pointer of the first record which fulfills conditions is found, the count field of the argument and record of the continuing SEARCHID command is compared, and if in agreement, orientation will be established on the record. If not in agreement, the comparison with the count field of the following record is repeated, orientation is established on the record which was in agreement with the argument first, and it moves to the following command (refer to a "SEARCH IDEQ [4.]" command about the case where the record which searches and is in agreement to the last of the logic truck of CKD is not found). In addition, although you may progress to the following command immediately when hit checking by the SEEK command to precede shows that the purpose truck already exists on a cache (at the time of a hit) Even if it has hit, when some trucks exist on a

cache and the time of a mistake hit or the frame in which the purpose record is contained does not exist on a cache After returning only a channel end CE, once disconnecting from a channel and carrying out the staging of the target frame from a physical disk, the device end DE will be returned and it will progress to the following command.

[0095] 2. SEARCH IDENTIFIRE 5 bytes of SEARCH information is compared with CCHHR of the count field (the R0 field is also included) read from reception and a device from an EQUAL <function> channel. <Implementation method> OSET By actuation shown by "1. SET SECTOR" when it chained from the SECTOR command, orientation is established on the record which corresponded. With an actual CKD disk, it is SET. After SECTOR, if CCHHR of the record found to the beginning is not in agreement, an inequality report is carried out (CE, DE), a host uses the transfer in channel TIC together, and it is SEARCH again. ID It waits to publish the EQ command. However, as "1. SET SECTOR" showed in this subsystem, it is 1 time of SEARCH. ID The whole truck will be able to be searched by the EQ command. If it says from the strict nature of an emulation, it will be one SEARCH to the last. ID Although the EQ command should perform only the search of one record, since especially the merit by doing so cannot be considered, I think that the method of a package search shown by "1. SET SECTOR" from the point of processing effectiveness may be used. SET If the multi-truck bit is set when there is no record which is in agreement with back from the sector specified by SECTOR, an inequality report (CE, DE) will be performed. However, when the record itself does not exist, a head switch is carried out on the next truck as it is. SET It is No, if the record which returns to the record (R0 is included) of the beginning of the frame of the beginning of the same truck, repeats a comparison, and is still in agreement is not found when the purpose record is not found in back from the sector specified by the SECTOR command (it contains also when a record does not exist) and a multi-truck bit is not set, either. Record Found is reported.

O READ, WRITE, a SEARCH system command, or SPACE The command preceded when it chains from the COUNT command compares CCHHR to the following record which was carrying out orientation. In this case, since orientation is already established, it is SET. The one record as follows of the record which is searched like the CKD disk of thing unlike the case where it chains from the SECTOR command is better. When the record does not exist more back than the place which was carrying out orientation, if the multi-truck bit is set, and it moves to the next truck and is not set, it returns to the head of the same truck and compares with it. However, it is No when saying to the last of a truck once again within the same CCW chain. RecordFound is reported.

O others -- although search actuation is performed to the record discovered first by the actual CKD disk in this case on a current truck when it is the head of the chain from a command, or a chain, and also it always searchs in an order from R0 in this subsystem -- "the case where it chains from a SET SECTOR command" -- being the same . When the mistake hit is carried out and the staging was not completed, or when the multi-truck bit has left and the next truck is carrying out the mistake hit, a staging is made to complete by this command. Moreover, the access truck of the last to the volume must always be kept in mind for recognition of the present truck when the SEEK command does not precede.

[0096] 3. READ The information on the data field read from the DATA <function> disk is transmitted to a channel.

<Implementation method> OREAD, WRITE, a SEARCH system command, or SPACE When it chains from the COUNT command, the information on the first data field which exists after the field which orientation had established on the frame is transmitted to a channel, and orientation is maintained. More back than the field which orientation had established, if the multi-truck bit is set when a data field does not exist, the data field of R1 (the next command of R0) of the next truck will already be transmitted. If the multi-truck bit is not

set, a head switch is not carried out but the data field of R1 of the same truck is transmitted (when transmitting the data field of R0, it is only the case where it chains from the Search system command with which were satisfied of the comparison of the R0 field, or the SPACE COUNT command which bypassed R0C).
O others -- since it is in the condition which orientation has not established when it is the head of the chain from a command, or a chain, it is not fixed which record an actual CKD disk also detects. Therefore, the information on the data field of R1 (the next command of R0) is especially transmitted, and it is satisfactory each time (if R1 exists). After a transfer establishes orientation as it is. If R1 does not exist and the multi-truck bit is set, the data field of R1 of the next truck will be transmitted (while the address mark has not been found, it corresponds, when the index has been detected). It is No if the multi-truck bit is not set. Record Found is reported. Of course, first, when the next truck does not exist on a cache, after carrying out a staging, data transfer is carried out to a channel. It is RD to the truck which accessed the last in the same volume when the SEEK command did not precede within the same CCW chain. The DATA command is executed.
[0097] 4. WRITE COUNT KEY AND The information received from the DATA <function> channel is written in on a disk as the Rn field (except for the R0 field).
<Implementation method> This command is SEARCH which corresponded. ID EQ command (RD D, RD KD, WR D, or WR KD may continue), SEARCH all whose keys corresponded KEY EQ command (RD D or WRD may continue), WR R0 command or WR Since it becomes command rejection except when it chains from the CKD command, when this command is executed, orientation has surely been established already as a matter of fact. Therefore, if information is received from a channel, the following operation will be performed immediately (refer to drawing 25).
(1) ** which gave other information on the count field of Rn-1 0 for CCHHRKLDL of Rn to reception and it from the receipt channel of the count field. Yet, it does not write out on a cache. As other information on the count field of Rn, they are SC (skip control), SN (segment number), PA (physical address), F (flag), etc. Since SC does not support skip control in this subsystem, it only writes constant value as a matter of fact (however, the secured direction feels that there is likely to be a use of the future something only for area). It calculates and writes in from the value of SC of the record which also precedes SN (refer to the examination of the check method of track capacity which used "segment number" for the count approach). Moreover, a sector value is also calculated from this value to coincidence, and it writes in the record pointer in a control frame by (3). PA and F are good with the same value as HA.
(2) from KL and DL which were received from the check channel of track overrun, and a segment number -- since -- investigate whether track overrun is generated (refer to the examination of the check method of track capacity which used "segment number" for the detailed count approach). When generating track overrun, even the track overrun point performs data transfer by subsequent control, and control which reports track overrun to a host is performed.
(3) If the check of the additional track overrun of a record pointer finishes, the memory address which shows the head of the control frame in the control (CPS holds by itself) frame pointer of the frame which has established current orientation will be 4B Reduced (4B escape of a control frame is done), and the record pointer of Rn will be further added to the head of a control (CPS holds) frame. When a control frame eats into a record frame by 4B Reducing, adding a record to the frame gives up, and it adds a record pointer to the control frame of the following frame. If required, I will have the area of the following frame secured to a manager.
(4) Calculate the memory address at the tail end of each field of KL and DL to Rn which were received from

the check channel of eye a ** paddle of a frame, and compare with the memory address in the control frame pointer updated by (3). When the memory address at the tail end of the field of one of Rn is eating into the control frame, after carrying out an end halt of the data transfer with a host before a control frame and advancing a memory address to the head of the following frame, control which resumes data transfer with a host is performed (CPS). If the tail end of Rn is settled in the record frame, the above control is unnecessary, and it can be performed, without interrupting the data transfer of Rn on the way.
(5) the beginning to the cache of the count field -- after a series of checks of these are completed, write out the data of the count field in CPS to the applicable address on cache memory. When eye a ** paddle of a frame comes in the count field, the beginning is resumed, after once stopping the beginning to a cache there and advancing a memory address to the head of the following frame.
(6) After suitable gap equivalent time progress, also receive the data of a data field from a channel and write them in a cache, after receiving the data of key field from a channel and writing in on a cache first, if the tail end of Rn is settled in the record frame as a result of the check of the receipt (4) of key field and a data field. If the check of (4) shows encountering the break of a frame on the way, data transfer will once be interrupted there. Since CPS also knows the address on the cache of the following frame when the frame which current orientation has established is not the last frame, after advancing the memory address for data transfer to the head of the following frame, the writing to data transfer/cache with a channel is resumed. When the frame which has established current orientation is the last frame Since the field of the frame after the degree is not secured on the cache from the first After CPS's having required the field for the following frame of the manager and having the address taught from a manager, Data transfer is resumed to there (what is necessary is for the treatment of partitioning of such a new frame etc. to be the timing of a gap, and just to perform it, while having interrupted data transfer). To the frame which carried out data transfer, the update flag of the FBA block with which it corresponds in the control frame of each frame is set.
(7) the report of CE -- if it ends to here, CPS will report the channel end CE which shows that the data transfer between channels was once completed to the channel in order to check the existence of a command chain.
--- When there are no command chain directions, he is a (8) CPS-> manager. If there are no destaging demand command chain directions, it will also set the update flag of the FBA block corresponding to the part which carried out zero padding to the remaining part of the frame to which the tail end of Rn was written while carrying out zero padding of the CPS. Subsequently, after writing out the control frame of the updated frame on a cache, destaging is required of DPS through a manager. Under the present circumstances, in the last frame ****** case back more from the first than the frame to which the tail end of Rn was written, CPS also reports that the frame after the frame at the tail end of Rn became an invalid to a manager. It does not report that subsequent frames became an invalid from the frame to which the tail end of Rn was written from the first anew since the frame peach after it was an invalid frame when the last frame was a front (it contains also when the same).
(9) Manager -> DPS When it is also united and reported that the frame which became an invalid exists, a destaging demand manager directs the destaging of an applicable truck to DPS, after deleting the invalid frame on the cache of an applicable truck (its management information is maintained). Under the present circumstances, while a manager tells the address of a frame with which the effective data on a cache exists to DPS as the address of the data which should be written out to a physical disk Only when an invalid frame newly occurs, as data written in an invalid frame The data for invalid frames are beforehand prepared for one somewhere, and the address there is directed (that is, all invalid frames will write the same data on the memory for which the data for invalid frames are prepared in a physical disk.). A manager may prepare this

data for invalid frames on cache memory, and each DPS may have it in its own board. When an invalid frame does not newly occur, rewriting anew the frame which was an invalid frame from the first to DPS does not direct.
(10) Destaging DPS receives the directions from a manager and writes out the data on cache memory to a physical disk. DPS performs a light about the FBA block the update flag of the frame directed by the manager has left.
(11) After the beginning to a destaging termination report physical disk is completed, DPS performs a termination report to CPS through a manager.
(12) The DE report CPS reports the device end DE which shows that all processings were completed to the channel.
--- If (8)' command chain is directed when there are command chain directions, CPS will report DE and will receive the following command.
(9) -- ' -- the following command -- WR If it is CKD or ERASE, it will move to the processing. If it is other commands, CPS will report CE, UCK, and SM and will require a command retry.
(10) 'above(8) - (11) is performed after this.
(11) If the light termination to a physical disk from 'DPS is reported, CPS will report DE and will move from the command which carried out the command retry demand previously to reception repair and activation of the command.
[0098] 5. WRITE The data field of a record is updated using the information received from the DATA <function> channel.
<Implementation method> This command is SEARCH which corresponded. ID SEARCH the EQ command or all whose keys corresponded KEY Since it becomes command rejection except when it chains from the EQ command, when this command is executed, orientation has surely been established already as a matter of fact. Therefore, if information is received from a channel, the following operation will be performed immediately.
(1) Calculate the memory address at the tail end of KL of the count field of the record which is going to write in the check data of eye a ** paddle of a frame, and DL to Rn, and compare with the memory address in the control frame pointer of the frame which has established current orientation. When the memory address at the tail end of Rn is eating into the control frame, after carrying out an end halt of the data transfer with a host before a control frame and advancing a memory address to the head of the following frame, control which resumes data transfer with a host is performed (CPS). If the tail end of Rn is settled in the record frame, the above control is unnecessary, and it can be performed, without interrupting the data transfer of Rn on the way.
(2) If the tail end of Rn is settled in the record frame as a result of the check of the receipt (1) of a data field, the data of a data field will be received from a channel and it will write in a cache. If the check of (1) shows encountering the break of a frame on the way, data transfer will once be interrupted there. Since the following frame surely exists in such a case in the case of an update light, CPS resumes the writing to data transfer/cache with a channel, after advancing the memory address for data transfer to the head of the following frame. To the frame which carried out data transfer, the update flag of the FBA block with which it corresponds in the control frame of each frame is set.
(3) the report of CE -- if it ends to here, CPS will report the channel end CE which shows that the data transfer between channels was once completed to the channel in order to check the existence of a command chain.
--- When there are no command chain directions, he is a (4) CPS-> manager. If there are no destaging

demand command chain directions, destaging will be required of DPS through a manager. An invalid frame seems not to newly generate in the case of an update light.
(5) Manager -> DPS A destaging demand manager tells the address of a frame with which the effective data on a cache exists to DPS as the address of the data which should be written out to a physical disk, and directs destaging.
(6) Destaging DPS receives the directions from a manager and writes out the data on cache memory to a physical disk. DPS rewrites only the FBA block which has an update flag among the frames directed by the manager.
(7) After the beginning to a destaging termination report physical disk is completed, DPS performs a termination report to CPS through a manager.
(8) The DE report CPS reports the device end DE which shows that all processings were completed to the channel.
--- If (4)' command chain is directed when there are command chain directions, CPS will report DE and will receive the following command.
(5) -- ' -- if the following command is the command of a SEARCH system, or a format light system command, it will move to the processing. Within the same command chain, when commands other than a SEARCH system or a WRITE system are directed, CPS reports CE, UCK, and SM and requires a command retry.
(6) 'above(4) - (7) is performed after this.
(7) If the light termination to a physical disk from 'DPS is reported, CPS will report DE and will move from the command which carried out the command retry demand previously to reception repair and activation of the command.
[0099] As mentioned above, according to this example, all gaps, such as an inter-record gap and a gap between the fields, are deleted from one truck of a variable-length record. The truck data is divided into the management unit equivalent to the integral multiple of the fixed-block-length size of fixed-length record. After writing the control information which shows the location of all the variable-length records contained in the management unit in an order from back in the tail end of the above-mentioned management unit, while dividing into the fixed-block-length size of fixed-length record Since the information which shows the relative position from the head of the truck in a variable-length record format to each fixed-length record is made to hold While there are few counts of memory access for the localization of the target variable-length record, they end within a fixed block and becoming memory unloading It is suitable for the disk subsystem which there was little memory space for holding truck data, and ended since all the gaps were deleted, therefore used the disk cache as the base, and the hit ratio of a cache improves. Furthermore, size of a control information field is made to adjustable, and that what is necessary is to prepare only the number of a variable-length record, there is little memory and it ends.
[0100] Although the case where gave behind a control frame and the die length of a control frame was made adjustable was shown as shown in drawing 25 , you may make it establish a positional information field, i.e., a control frame, in the head of each management unit in the example 2. above-mentioned example 1, as shown in drawing 24 . However, since it is necessary to set up the positional information field for 94 records beforehand as mentioned above in this case, when one management unit is set to 12KB, the lack of capacity will arise. Therefore, it is necessary to set one management unit to 13KB. Thus, since the area of a record frame is also fixed while the size of a control frame is fixed by having a control frame beforehand by Bx4 94 records, the access approach of a management unit becomes simple.
[0101] In the example 3. above-mentioned example 1, although the case where a sector value was used

was shown as a relative position from the head of the truck in a variable-length record format, since the relative position used by the set sector command is a sector value, this is because it is desirable to use a sector value. When the set sector command uses the segment value from the head of a truck, other relative positions, for example, variable-length record format, this relative position may use not a sector value but a segment value. That is, the value of the sector value of the record pointer shown in drawing 25 may be a segment value.

[0102] In the example 4. above-mentioned example 1, when track capacity was calculated, the case where the segment value was given to the CKD records of each was shown, but when managing it per sector rather than managing track capacity per segment, not a segment value but a sector value may be given.

[0103] In the example 5. above-mentioned example 1, although the case where gave a sector value to the record pointer in a control frame, and a segment value was given in each CKD record in a record frame was shown, the case where have a fixed relation (sector value x7= segment value), and either is used is sufficient as a sector value and a segment value. That is, if it has the sector value to the record pointer, it is not necessary to keep a segment value in mind on each CKD record. It becomes impossible however, for an operating capacity of a truck to calculate only a sector unit in this case. On the other hand, when it does not have a sector value to a record pointer but the segment value is given to the CKD record in a record frame, it enables an operating capacity of a truck to calculate per segment. Since a sector value does not exist in a record pointer, a CKD record is discovered from the memory address of a record pointer, and if a sector value is not computed from the segment value in that CKD record, it becomes impossible however, for whether it is the sector currently searched to judge in this case, when a sector value is specified by the set sector command.

[0104] Although not clearly written especially in the example 6. above-mentioned example 1, when the FBA disk 9 consists of one disk, which [in the case of consisting of array form disks] case is sufficient. Moreover, although not clearly written in the above-mentioned example 1, as long as the FBA disk 9 is equipment which can record data, such as a magnetic disk drive or an optical disk unit, it may use what kind of medium.

[0105] Although the case where a CKD-FBA conversion method existed in disk storage control 5 in the example 7. above-mentioned example 1 was shown, the channel of not only when it is disk storage control, but the host computer 1 may perform performing CKD-FBA conversion. Or the storage itself, such as a magnetic disk drive, may carry out. Drawing 32 is drawing showing one example in case the storage itself, such as a magnetic disk drive, performs CKD-FBA conversion. As shown in drawing 32 , the variable-length record memorized by the cache memory of the disk storage control 500 of CKD is changed, and in rememorizing on the FBA disk 900 with a fixed length block, the CKD-FBA transducer 100 in the FBA disk 900 operates. The gap cutout 101 reads into cache memory 600 the variable-length record memorized by the cache memory of disk storage control 500. In this case, all the records of one truck of the cache memory of disk storage control 500 are read into cache memory 600. And an inter-record gap and the gap between the fields are deleted from the variable-length record for this one truck. The truck image 700 of CKD read into cache memory 600 is changed into truck data 7a from which the gap was deleted at this time.

[0106] Next, the record layout section 102 arranges truck data 7a in order in the management units 7b, 7c, 7d, and 7e called a frame. The size of the management units 7b-7e called this frame has the size of 12KB equally divided into four, respectively, when the truck image 7 of CKD has the size which is 48KB. The record layout section 102 arranges the variable-length record of truck data 7a from frame 7b in order. In this case, the address information which shows the location of the variable-length record contained in each

frame is memorized on each frame. Moreover, the sector information which shows the relative position (sector value) from the truck head where the record was memorized by the CKD disk 10 is memorized similarly.

[0107] Next, the fixed block storage section 103 memorizes a frame in the storage section 200 of the FBA disk 900. If size of the fixed length block of the FBA disk 9 is set to 1KB here, respectively, since Frames 7b, 7c, 7d, and 7e have the size of 12KB, respectively, one frame will be recorded using 12 fixed length blocks.

[0108] Thus, the data for one truck outputted from the disk storage control of CKD are memorized by the storage section of FBA. On the contrary, reading from the storage section 200 of the FBA disk 100 is attained by performing the reverse of the above-mentioned actuation.

[0109]

[Effect of the Invention] In this invention, the memory space which holds truck data by deletion of the above-mentioned gap can be reduced. Moreover, since the localization of the CKD record made into the purpose by the above-mentioned positional information is made, it is useful to reduction of the count of memory access, and memory space.

[Translation done.]

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（Ａ）**　(11)特許出願公開番号

特開平5－307440

(43)公開日　平成５年(1993)11月19日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　3/06	３０１　Ｖ	7165－5B		
	Ｙ	7165－5B		
Ｇ１１Ｂ　20/12		7033－5D		

審査請求　未請求　請求項の数11(全 57 頁)

(21)出願番号　特願平5－9002

(22)出願日　平成５年(1993)１月22日

(31)優先権主張番号　特願平4－49833

(32)優先日　平４(1992)３月６日

(33)優先権主張国　日本（ＪＰ）

(71)出願人　000006013
三菱電機株式会社
東京都千代田区丸の内二丁目２番３号

(72)発明者　中村　洋一
鎌倉市上町屋325番地　三菱電機株式会社
コンピュータ製作所内

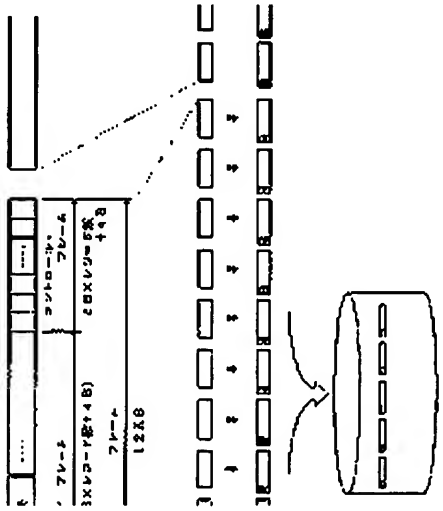
(74)代理人　弁理士　高田　守

(54)【発明の名称】　データ記憶フォーマット変換方式及びその変換方法及びアクセス制御装置及びデータアクセス方法

(57)【要約】

【目的】　ＣＫＤレコードをＦＢＡレコードにフォーマット変換する際、目的とするＣＫＤレコードを発見するためのメモリアクセス回数が少なくてすみメモリ負荷軽減がなされると共に、トラックデータを保持するためのメモリ容量が少なくてすむレコード記憶フォーマット変換方式を得る。

【構成】　ＣＫＤレコードの１トラックからレコード間ギャップ及びフィールド間ギャップ等全てのギャップを削除し、そのトラックデータをＦＢＡレコードの固定ブロック長サイズの整数倍に相当する管理単位であるフレームに分割し、その管理単位に含まれる全てのＣＫＤレコードの位置を示す制御情報をコントロールフレームと



1

【特許請求の範囲】

【請求項1】　可変長フォーマットで記憶されたレコードを固定長フォーマットに変換して記憶するデータ記憶フォーマット変換方式において、

可変長フォーマットで記録される1トラック分の可変長レコードからレコード間ギャップ及びフィールド間ギャップを削除してトラックデータを生成するギャップ削除手段、

上記ギャップ削除手段によりギャップを削除したトラックデータを所定のサイズの管理単位に配置し、その管理単位に含まれる可変長レコードの位置を示す第1の位置情報と、その管理単位に含まれる可変長レコードの可変長レコードフォーマットでのトラックの先頭からの相対位置を示す第2の位置情報とを上記管理単位に記憶するレコード配置手段、

上記管理単位を固定長レコードの固定長ブロック長サイズに分割して記憶する固定ブロック記憶手段を備えたことを特徴とするデータ記憶フォーマット変換方式。

【請求項2】　上記レコード配置手段は、可変長レコードを上記管理単位の一方から順に配置するとともに、配置した可変長レコードに対応する第1と第2の位置情報を上記管理単位の他方から順に配置することを特徴とする請求項1記載のデータ記憶フォーマット変換方式。

【請求項3】　上記レコード配置手段は、あらかじめ管理単位内に、第1と第2の位置情報を記憶する領域と可変長レコードを記憶する領域とを定めておくことを特徴とする請求項1記載のデータ記憶フォーマット変換方式。

【請求項4】　上記第1の位置情報は管理単位に含まれる可変長レコードの管理単以内のメモリアドレスであり、上記第2の位置情報はトラックの先頭からのセクタ値、あるいはセグメント値であることを特徴とする請求項1記載のデータ記憶フォーマット変換方式。

【請求項5】　上記可変長レコードは、可変長レコード内に、可変長レコードフォーマットでのトラックの先頭からの相対位置を示すセグメント情報を有していることを特徴とする請求項1記載のデータ記憶フォーマット変換方式。

【請求項6】　可変長フォーマットからギャップを削除し、そのデータを所定サイズの管理単位に配置し、その管理単位に含まれる可変長レコードの位置を示すアドレス情報と、その管理単位に含まれる可変長レコードの可変長フォーマットでの先頭からの相対位置を示す相対位

2

管理単位ごとに格納するメモリ、

（c）レコードが可変長フォーマットで記憶されているものとして出力された可変長レコードへのアクセス命令を入力する入力手段、

（d）上記入力手段により入力したアクセス命令に基づいて、アクセスする可変長レコードが記憶されていると考えられる管理単位の位置を概算する位置計算手段、

（e）上記位置計算手段により概算された位置にある管理単位を上記読み書き手段により上記記憶部から上記メモリに読み込み、上記入力手段により入力されたアクセス命令がアクセスしようとするレコードを検索するレコード検索手段。

【請求項7】　上記アクセス制御装置は、さらに、相対位置情報を用いて、可変長レコード方式でレコードが記憶されている場合のトラックの使用容量を判定することを特徴とする請求項6記載のアクセス制御装置。

【請求項8】　上記入力手段はアクセスしようとするレコードの相対位置情報を入力するとともに、

上記レコード検索手段は、

上記アクセスしようとするレコードの相対位置情報を、検索された管理単位に保持されている相対位置情報とを比較し、アクセスしようとするレコードの相対位置情報を持つ管理単位が見つかるまで、読み書き手段により次の管理単位を記憶部からメモリに読み込むことにより管理単位を検索する管理単位検索手段と、上記管理単位検索手段によりアクセスしようとするレコードの相対位置情報を持つ管理単位が見つかった場合に、その管理単位に含まれるアドレス情報を用いてアクセスしようとする可変長レコードを特定するレコード特定手段を有することを特徴とする請求項6記載のアクセス制御装置。

【請求項9】　以下の工程を有するデータ記憶フォーマット変換方法

（a）データを有するフィールド及びレコード間ギャップ及びフィールド間ギャップを有する可変長フォーマットからレコード間ギャップ及びフィールド間ギャップを削除して各可変長レコードのフィールドを連結するギャップ削除工程、

（b）上記ギャップ削除工程により連結された各可変長レコードをあらかじめ定められたサイズを持つフレームに順に配置するとともに、各可変長レコードをアクセスするためのアクセス情報を各可変長レコードに対応させて記憶するレコード配置工程、

（c）上記フレームを複数の固定長のブロックサイズに

方法。

【請求項１１】　以下の工程を有し．上記請求項９記載のデータ記憶フォーマット変換方法により記憶された固定長フォーマットのデータをアクセスするデータアクセス方法

（ａ）可変長フォーマットのレコードをアクセスするためのコマンドとアクセスするデータの検索情報を入力するアクセスコマンド入力工程、

（ｂ）上記アクセスコマンド入力工程により入力した検索情報からアクセスするレコードが記憶されているフレームの位置情報を推定する位置情報推定工程、

（ｃ）上記位置情報推定工程により推定された位置情報をもつフレーム以降のフレームを順に固定長フォーマットで記憶された複数の固定長ブロックから再生し．アクセスするレコードを検索するレコード検索工程。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】この発明は、たとえば、汎用計算機の磁気ディスク装置で用いられている可変長レコード方式を市販の小型ディスク装置で用いられている固定長方式で実現するためのデータ記憶フォーマット変換方式等に関する。

【０００２】

【従来の技術】従来、汎用計算機システムの外部記憶装置としては、可変長レコード（ＣＫＤ；Ｃｏｕｎｔ　Ｋｅｙ　Ｄａｔａ）方式の固定ディスク装置が広く用いられている。可変長レコード方式とは．磁気ディスク上の１トラックに記録する物理的なレコードの長さも数も可変である記録フォーマットである。一般的に可変長レコード方式の磁気ディスクの１トラックは図３３に示すようにインデックス・マーク（図示せず）から始まり、まず最初にホーム・アドレス（ＨＡ）と引き続くレコード・ゼロ（Ｒ０）が存在する。このＨＡとＲ０はそのトラックのシリンダ・アドレスやヘッド・アドレスなどのアドレス情報以外に、そのトラック上の磁気的な傷の位置や．不良トラックであるか否かの情報．また、不良トラックであれば．交代割り付けされている相手先のトラックのアドレス等の制御情報を含んでいるため、必ず存在しなくてはならない。このＨＡ、Ｒ０に引き続くレコード１（Ｒ１）以降のレコードが通常のデータを記録するレコードで、レコード１個１個の長さも個数もその総容量がトラックに記録できる容量である限り、ホスト計算機上のソフトウェア（Ｓ／Ｗ）から自由にフォーマット

ス情報のほかに．そのレコードのキー・フィールドの長さ（ＫＬ）、データ・フィールドの長さ（ＤＬ）などを含んでいるので．引き続くキー・フィールド、データ・フィールドの長さを自由に変えることができる。キー・フィールド、データ・フィールドがそのレコードの実際のデータを記録する部分であるが、そのレコードのキーを記録すべきキー・フィールドが不要な場合は．カウント・フィールドでＫＬ＝０を指定することにより．キー・フィールドの存在しないレコードを作ることもできる。ＨＡ、Ｒ０．Ｒ１、Ｒ２・・・などの間には．適当なサイズのレコード間ギャップ（Ｇ１．Ｇ２，Ｇ３）が設けられており．特にＲ１、Ｒ２・・・などのレコードの前のギャップ（Ｇ３）には、アドレス・マーク（ＡＭ）と呼ばれる磁気的に特別なマークが記録されているため．トラック内のどの部分から読み始めても．このアドレス・マークを検出することにより．レコードの読み始めを見つけることができる。

【０００３】また、各レコードの内部でも各フィールドを区別するためにそれぞれフィールド間ギャップ（Ｇ２）が設けられている。これに対し．ミニコンピュータ．オフィスコンピュータ．パーソナルコンピュータ、ワークステーションなどの比較的小規模な計算機システムの外部記憶装置としては．固定長レコード（ＦＢＡ；Ｆｉｘｅｄ　Ｂｌｏｃｋ　Ａｒｃｈｉｔｅｃｔｕｒｅ）方式の固定ディスク装置が用いられている。固定長レコード方式のディスクでは．通常一つのレコード（固定長レコード方式のディスクではセクタと呼ばれることが多い）はアドレス情報を記録しているＩＤフィールドとデータを記録するデータ・フィールドからなりたっており．いずれのフィールドの長さも固定である。従って、１トラックあたりのレコード数（セクタ数）も固定である。

【０００４】従来は、可変長レコード方式の磁気ディスクと固定長レコード方式の磁気ディスクは、対象とする計算機システムによって上述のような使い分けが行われてきた。従って．固定長レコード方式のディスクは、対象とするシステムが小さいことから．可変長レコード方式のディスクに比べ、一般的に小型で記憶容量も小さく．データ転送速度、シーク速度などの性能も低かった。しかし、近年、可変長レコード方式のディスクも小型化に対する要求が著しく高くなってきたこと．逆に、固定長レコード方式のディスクの方は．大容量化．高速化が著しいことなどから　什様面での両者の差異は小さ

方式のディスクを利用したいと言う要求が強まっている。

【０００５】また、ＣＫＤ方式の固定ディスク装置では、データを記録するデータ面と、ヘッドを位置決めするためのサーボ情報を記録するサーボ面を別々に設けるサーボ面サーボ方式が用いられてきた。この方式では、データ面とサーボ面の機械的な位置の精度によってヘッドの位置決め精度も規定されてしまう。そこで、さらにトラックの高密度化を図るため、データ面中のＩＤフィールドなどにサーボ情報も書き込んでしまうデータ面サーボ方式が用いられるようになってきた。このデータ面サーボ方式を用いると、極端には、１トラック１レコードも許される可変長レコード方式では常に一定間隔でサーボ情報が得られるという保障はなく、必ず一定間隔でサーボ情報が得られる固定長レコード方式に比べ、ヘッドの位置決め精度に関して不利になる。さらに、ディスク1台1台の性能向上や信頼性向上には限界が有るため、複数のディスクにデータを分散して記録したり、その分散したデータ間のパリティなどを記録する冗長ディスクを設けるなどして、性能・信頼性の向上をはかるディスク・アレイの技術が注目されているが、レコードの長さや個数が可変である可変長レコード方式のままでは、このような技術に対応することも困難である。しかし、汎用計算機の世界では膨大なソフトウェア資産がきづかれているため、ディスクの記録フォーマットを可変長から固定長に変更することは著しく困難である。

【０００６】ここで、図３４及び図３５を用いて従来の可変長レコード方式で記録された可変長レコードへのアクセス方式について述べる。図３４はサーボ面サーボ方式が用いられている磁気ディスクを模式的に示す図であり、この磁気ディスク装置にはサーボ面Ｓとデータ面Ｄが設けられている。サーボ面Ｓにはセクタ（ＦＢＡ方式のセクタとは異なる概念のセクタ）が論理的に設けられている。図３５はこのセクタを説明するための図である。ここで１トラックの記憶容量を４８ＫＢであるとするとこの１トラックは複数のセクタに分割される。ここでは１セクタのサイズを２４４Ｂとする。この１セクタはさらにセグメントＳ１～Ｓ７という７つのセグメントから構成されている。１セクタのサイズが２４４Ｂであるので１セグメントは３２Ｂになる。このような磁気ディスク装置を用いてデータ面Ｄにデータを記録する場合には、データが記録されたセクタの値を記憶しておき、そのデータを検索する場合には、データ面に書かれたとにより指定する。このセットセクタコマンドにより磁気ディスク装置は、ホスト計算機及び磁気ディスク制御装置から切り離され、指定されたセクタを自律的に検索する。指定されたセクタを発見すると磁気ディスク装置は、磁気ディスク制御装置を介し、ホスト計算機に再接続要求を出し、再接続が行なわれる。次にサーチＩＤコマンドによりサーチすべきデータのＩＤが送られてくる。このＩＤは、例えばシリンダ番号やトラック番号や、あるいはサーチすべきレコード番号である。このサーチＩＤコマンドを受け取ると磁気ディスク装置は、そのＩＤとセットセクタコマンドにより指定されたセクタから最初に存在するレコードのＩＤを比較し、その結果をホスト計算機に報告する。両者のＩＤが一致していなければ、コマンドの実行は次のトランスファーインチャネルコマンドにより、セットセクタコマンドにジャンプして戻され、次のレコードに対し、再度サーチＩＤコマンドが発行される。両者のＩＤが一致していれば、ホスト計算機は次のトランスファーインチャネルコマンドをとばし、最後のリードデータコマンドあるいはライトデータコマンドを発行することにより、検索したデータに対し、リードまたはライトという命令を実行する。

【０００７】このように汎用計算機で用いられるソフトウェアは既に可変長レコード方式に基づいて作成されており、既に築かれたソフトウェアを変更するということは著しく困難である。そこで、実際のディスクは固定長レコード方式のディスクを使用しながら、磁気ディスク制御装置で可変長レコード方式をエミュレーションすることにより、ホスト計算機上のソフトウェアからは従来どおりの可変長レコード方式のディスクを使用できる技術が考えられている。

【０００８】ＣＫＤ－ＦＢＡ変換については「記憶制御方法及び装置」（特開平１－３０６９１号公報）において詳細に述べられている。また、ＣＫＤ－ＦＢＡ変換については、さらにＩＢＭ社の出版物「ＩＢＭ４３２１／４３３１プロセッサ互換性機能」（ＧＡ３３－１５２８，第３版、１９８２年９月）に一般的に記載されている。これらの文献の内容をまとめると、下記のようになる。上記の出版物ではＣＫＤからＦＢＡに変換する際すべてのギャップを取って圧縮している。また、上記の出版物では図３６に示したセットセクタコマンドがサポートされていないことが記述されている。上記の出版物ではセットセクタコマンドが出された場合には、ノーオペレーションになると記載されている。セットセクタコマ

近のセクタを指定するものであり、セットセクタコマンドで指定されたセクタ以後をサーチＩＤコマンドによりサーチする場合に比べてアクセス時間の点で非常に不利である。

【０００９】一方、上記公報では（カッコ内の番号は図３７に対応）．

（１）フィルード間ギャップ（Ｇ２）やＥＣＣ．パディング．物理的パラメータなどは取る。

（２）レコード間ギャップ（１３１．１３３，１３７，１６３）は残す。

（３）（１）で取ったフィールド間ギャップなどの分は．同じ分だけ（２）のレコード間ギャップを伸ばす。これにより、トラックの先頭から各レコードのカウント・フィールドまでの相対位置は元のまま維持される。（キー・フィールドやデータ・フィールドの相対位置はずれることになるが、カウント・フィールドの相対位置さえ分かれば．レコードの位置決めには差し支えない）。

（４）このトラック・データをＦＢＡの固定ブロック（セクタ）に分割し、その際、各固定ブロックに含まれる最初のレコードのカウント・フィールドの位置情報（１５６）を各固定ブロックの先頭に付加しておく。これにより、ＦＢＡ方式のディスクに記憶されたＣＫＤレコードへの直接かつランダムなアドレッシングが可能になる。

（５）そのブロックにカウント・フィールドが含まれない場合は、それを示す標識ビットを記録しておく。

さらに整理すると、

（１）各レコードのカウント・フィルードの相対位置を維持するためレコード間ギャップだけを残し、他の余分なギャップなどは取り除く。

（２）ＦＢＡの各固定ブロックの先頭に、そこに記録されている最初のカウント・フィールドの位置を示すヘッダを付ける。

の２点が要件となって、こうすることによりＦＢＡ方式のディスクに記録したＣＫＤレコードへの直接かつランダムなアドレッシングを可能にする。

【００１０】

【発明が解決しようとする課題】しかしながら．上述した先行技術においては次の問題点がある。前述した出版物による技術によればＣＫＤからＦＢＡに変換する際、全てのギャップをとって圧縮しているがセットセクタコマンドがサポートされていないためにレコードのアクセス

コード方式のレコード）に含まれるＣＫＤレコードのうち先頭のＣＫＤレコードの位置に関する情報しか持たないので、先頭以外のＣＫＤレコードを見つけるためには．まず先頭のＣＫＤレコードのカウント・フィールドを読んで、その中のＫＬ．ＤＬから先頭のレコード長を知り．次のレコードのカウント・フィールドの位置を計算してから、また、次のレコードのカウント・フィールドのＫＬ，ＤＬを読んで．また、その次のレコードのカウント・フィールドを探すという手順を繰り返さなくてはならない。従って、処理時間がかかるばかりでなく、トラック・イメージがディスク・キャッシュ上などに集中管理されている場合は．そのメモリへのアクセス回数が著しく増え．システム全体の性能を低下させる。さらに．ＣＫＤレコードの位置情報を各ＦＢＡブロック毎という小さい単位に持つためチャネルとのデータ転送時、この位置情報をとび越すためのデータ転送一時中断が頻繁に発生し、データ転送効率の悪化を招く。

【００１２】この発明は．上述した従来例における問題点を解消するためになされたもので．ＣＫＤレコードをＦＢＡレコードにフォーマット変換する際、目的とするＣＫＤレコードを発見するためのメモリアクセス回数が少なくてすみメモリ負荷軽減がなされると共に．トラックデータを保持するためのメモリ容量が少なくてすむレコード記憶フォーマット変換方式を得ることを目的とする。

【００１３】

【課題を解決するための手段】この発明に係るデータ記憶フォーマット変換方式は．汎用計算機システムの外部記憶装置に用いられる記憶フォーマットである可変長レコードの１トラックからレコード間ギャップ及びフィールド間ギャップ等全てのギャップを削除し、そのトラックデータを固定長レコードの固定長ブロックサイズの整数倍に相当する管理単位に分割し、その管理単位に含まれる全ての可変長レコードの位置を示すアドレス情報と各可変長レコードの可変長レコードフォーマットでのトラックの先頭からの相対位置を示す相対位置情報とを上記管理単位に書き入れた後．固定長レコードの固定長ブロックサイズに分割して記憶することを特徴とするものである。

【００１４】また、この発明に係るアクセス制御装置は前述したデータ記憶フォーマット変換方式を用いて作成されたデータを記憶する記憶部に対してアクセスするアクセス制御装置であり．従来の可変長レコード方式に基

り．（ｃ）レコードが可変長フォーマットで記憶されているものとして出力された可変長レコードへのアクセス命令を入力する入力手段．（ｄ）上記入力手段により入力したアクセス命令から．アクセスする可変長レコードが記憶されていると考えられる管理単位の位置を概算する位置計算手段と、（ｅ）上記位置計算手段により概算された位置にある管理単位を上記読み書き手段により上記記憶部から上記メモリに読み込み．上記入力手段により入力されたアクセス命令がアクセスしようとするレコードを検索するレコード検索手段。

【００１５】また、この発明に係るデータ記憶フォーマット変換方法は可変長レコード方式により作成された可変長レコードからギャップを削除し．これをあらかじめ定められた管理単位であるフレームに配置すると共に配置したデータの位置等のアクセス情報を記憶させ．このフレームを固定長のブロックサイズに分割して固定長フォーマットとして記憶するものであり．以下の工程を有するものである。

（ａ）データを有するフィールド及びレコード間ギャップ及びフィールド間ギャップを有する可変長フォーマットからレコード間ギャップ及びフィールド間ギャップを削除して各可変長レコードのフィールドを連結するギャップ削除工程．（ｂ）上記ギャップ削除工程により連結された各可変長レコードをあらかじめ定められたサイズを持つフレームに順に配置するとともに、各可変長レコードをアクセスするためのアクセス情報を各可変長レコードに対応させて記憶するレコード配置工程、（ｃ）上記フレームを複数の固定長のブロックサイズに分割して固定長フォーマットとして記憶する固定長ブロック記憶工程。

【００１６】また、この発明に係るデータアクセス方法は前述したデータ記憶フォーマット変換方法により作成された固定長のブロックサイズに分割されたデータに対して従来から存在する可変長レコード方式のアクセス命令によりアクセスするための方法であり、以下の工程を有するものである。

（ａ）可変長フォーマットのレコードをアクセスするためのコマンドとアクセスするデータの検索情報を入力するアクセスコマンド入力工程、（ｂ）上記アクセスコマンド入力工程により入力した検索情報からアクセスするレコードが記憶されているフレームの位置情報を推定する位置情報推定工程、（ｃ）上記位置情報推定工程により推定された位置情報をもつフレーム以降のフレームを

場合に記憶容量が少なくて済む。またレコード配置手段が管理単位の中に可変長レコードの位置を示す第１の位置情報と、可変長レコードとして記録された場合の相対位置を示す第２の位置情報を記憶しているため．従来のソフトウェアから可変長レコード方式ののアクセス命令が来た場合、可変長レコード方式の相対位置を示す第２の位置情報を用いて管理単位を検索し．次に検索された管理単位の中の可変長レコードの位置を示す第１の位置情報を用いて目的とするレコードを検索することが可能になる。このためアクセス回数が最小限で済む。

【００１８】また、この発明におけるアクセス制御装置においては、位置計算手段が、従来のソフトウェアからの可変長レコード方式のアクセス命令に付随するレコードの相対位置情報から固定長レコード方式による管理単位の位置を大まかに計算するため、レコード検索手段が．この計算された位置情報をもとに固定長レコード方式で記録されたデータをアクセスすることができる。このようにこの発明に係るアクセス制御装置は位置計算手段とレコード検索手段を備えることにより、従来例で説明したようなセットセクタコマンドをサポートすることが可能になる。

【００１９】また、この発明におけるデータ記憶フォーマット変換方法はギャップ削除工程によりギャップを削除したのち、レコード配置工程により各可変長レコードをフレーム内に順に配置すると共にそのアクセス情報を共にフレームに記憶させる点に特徴がある。さらに、このフレームは固定長のブロックサイズに分割されて記憶されるためフレームを固定長レコード方式で記憶することができる。

【００２０】また、この発明におけるデータアクセス方法においては位置情報推定工程がアクセスコマンドにより入力された検索情報から固定長レコード方式に基づいて記憶される場合の位置情報を推定し．レコード検索工程によりこの推定された位置情報に基づいてフレームを検索するのでフレームを検索する回数すなわちメモリのアクセス回数を最小限に押さえることが可能になり、高速なアクセスが可能になる。

【００２１】

【実施例】

実施例１．図１はこの発明を説明するための概略ブロック図である。ホスト計算機１はチャネル２を有している。チャネル２は図に示すように複数のチャネル２ａ～２ｅを有している。例えばこのうちのひとつのチャネル

単位に読み書きされる。

【００２２】次に、ＣＫＤ－ＦＢＡ変換の概略について説明する。ホスト計算機１では従来のＣＫＤ方式で記録されたデータをアクセスするためのソフトウェアが動作する。このソフトウェアからはＣＫＤコマンド４がチャネル２ｂを介して磁気ディスク制御装置５に出力される。例えばＣＫＤコマンド４がライトコマンドである場合には、ＣＫＤフォーマットのレコード３は可変長レコード３ａ，３ｂ，３ｃとして出力される。磁気ディスク制御装置５はこの可変長レコード３ａ，３ｂ，３ｃをキャッシュメモリ６に記憶する。その際、キャッシュメモリにはＣＫＤのトラックイメージ７が構成される。すなわちＣＫＤのトラックイメージ７は可変長レコード３ａ，３ｂ，３ｃをトラックの先頭から順に配置したイメージとなる。従来ならばここでフィールド間ギャップ及びレコード間ギャップが生成されて付加されることになるが、この場合にはギャップは全て取り除かれる。磁気ディスク制御装置５はキャッシュメモリ６に作られたＣＫＤのトラックイメージ７をＦＢＡフォーマットのレコード８に変換する。ＦＢＡフォーマットのレコード８は固定長のブロック８ａ～８ｅによって構成されている。磁気ディスク制御装置５はこのＦＢＡフォーマットのレコード８をＦＢＡディスク９に書き込む場合にはＦＢＡコマンド１０を出力する。ブロック８ａ～８ｅはＦＢＡディスク９にあるブロック９ａ～９ｅに対応してそれぞれ記憶される。

【００２３】図２は、前述したＣＫＤ－ＦＢＡ変換の概略をさらに詳しく説明するためのブロック図である。ホスト計算機１、磁気ディスク制御装置５、キャッシュメモリ６、ＦＢＡディスク９は前述した図１と同様のものでありここではその説明を省略する。この磁気ディスク制御装置５の詳細な動作について図３、図４、図５を用いてさらに説明する。

【００２４】図３、図４、図５はレコード記憶フォーマット変換方式を説明するものである。先ず、図３（ａ）に示す元のＣＫＤトラックフォーマットに対し、図３（ｂ）に示すように、レコード間ギャップ、フィールド間ギャップ等全てのギャップＧ１，Ｇ２，Ｇ３を削除する。

【００２５】次に、図３（ｂ）に示すギャップ削除後の例えば４８ＫＢあるＣＫＤレコードの１トラックデータを図３（ｃ）に示す如く、後述する制御情報を含んで、ＦＢＡレコードの固定ブロック長の整数倍（レコード数

【００２７】ここで、図４（ｄ）においては、レコードの管理単位内の位置情報のサイズを可変にすることにより、位置情報を記憶する場所をあらかじめ確保する必要がなく位置情報の総容量を減らすことになっている。すなわち、可変長レコードは管理単位の先頭から記載され、位置情報は管理単位の後ろから順番に記憶されて行くため位置情報の数と可変長レコードの数は一致しており、両者の記憶領域が管理単位内でぶつかるまで記憶できることになる。このようにして実際に記録されているレコードの分だけしか位置情報領域を使用しないような制御を可能にする。しかし、位置情報領域のサイズを可変にすると、データ転送時やフォーマットライト時の制御が複雑なものとなり得るが、ここでは次のような方式を採用して改善を図っている。

【００２８】今、図４（ｄ）において、レコードの位置情報の管理単位をフレーム、実際にＣＫＤレコードを記録する領域をレコードフレーム、制御情報を記録する領域をコントロールフレーム、レコードの位置情報をレコードポインタと呼ぶ。実際に記録されているレコードの分だけしかレコードポインタを使用しない。これによりレコードポインタ領域のサイズは、管理単位のサイズを１２ＫＢとし最大９４個のＣＫＤレコードが記録されレコード１個当りの位置情報として４Ｂとすると、最大でも、１ＣＫＤトラック当り、

４Ｂ×９４＝２７６Ｂ

で済む。この４Ｂの位置情報は図４（ｄ）においてはレコードポインタとして示されており、レコードポインタは１Ｂのレコードナンバと１Ｂのセクタ値と２Ｂのメモリアドレスから構成されている。レコードナンバは記録される可変長レコードの番号を記録しておくものである。セクタ値は記録されるレコードがＣＫＤレコード方式で記録される場合のセクタ値を記憶するものである。メモリアドレスはフレームに記憶されるレコードのフレーム内のアドレスを記憶するものである。セクタ値はセットセクタコマンドにより指定されたセクタ値と比較されることにより利用される。同様にレコードナンバはサーチＩＤコマンドにより指定されたレコードナンバと比較されるために利用される。メモリアドレスはレコードフレームに記憶されている実際のレコードのアドレスを示すものである。セットセクタコマンド及びサーチＩＤコマンドによりレコードポインタのセクタ値とレコードナンバが比較され一致した場合にこのメモリアドレスを用いてレコードが読み出される。

る。また、コントロールフレーム内のレコードポインタも後ろから順番に並べていく。実際のレコードがレコードフレーム内で前にあるものほど、それに対応するレコードポインタはコントロールフレーム内で後ろにくる。これにより、フォーマットライト時の制御も無理がなくなる。

【００３１】次に、図４（ｄ）に示すように、ＣＫＤレコードの位置情報等付加後、１２ＫＢのレコードフレームを、図５（ｅ）に示す如く、ＦＢＡレコードのブロックサイズ例えば１０２４Ｂ（１ＫＢ）相当に分割し、さらに、図５（ｆ）に示す如く、各レコードのデータフィールドにｉＤフィールドをそれぞれ設けて、図５（ｇ）に示す如く、ＦＢＡレコード方式のディスクを得る。

【００３２】また、固定ブロック内に記録されているＣＫＤレコード１個ごとにＣＫＤフォーマットでのトラックの先頭からの相対位置を示す情報、例えばセグメントナンバを残しておく。このようにすることにより、各レコードの位置が分かれば、そのレコードに記録されている相対位置を示す情報から直ちに既に消費しているＣＫＤフォーマット上でのトラック容量が分かる。

【００３３】次に、図２に戻り磁気ディスク制御装置５の動作について説明する。磁気ディスク制御装置５はホスト計算機１とＦＢＡディスク９との間でデータの変換及び転送を行うチャネルパスサーバ５１（以下、ＣＰＳともいう）を有している。チャネルパスサーバ５１にはホスト計算機１からコマンド及びデータを入力するための入力部５２を有している。また、ＦＢＡディスク９からのデータをホスト計算機１に出力するための出力部５２を有している。入力部５２からホスト計算機１で動作しているソフトウェアによる可変長レコード方式のアクセス命令が入力された場合に位置計算部５３は例えばセットセクタコマンドで指定されたセクタ値に基づいてアクセスする可変長レコードが記憶されていると考えられる管理単位の位置を計算する。この位置計算部による計算は前述したようにＦＢＡディスク９にデータが記録される場合には全てのギャップが取り除かれているためにＣＫＤディスク１０に存在している場合よりも前方向に詰まった形でデータが記憶されており、位置計算部５３は推測により、アクセスしようとしてる可変長レコードが記憶されていると考えられる管理単位の位置を計算する。レコード検索部５４は位置計算部５３により計算された管理単位の位置をもとに管理単位をキャッシュメモリ６に読み込みアクセスしようとするレコードを検索する場合にアクセスしようとする可変長レコードをその管理単位に含まれるアドレス情報を用いて特定する。ＦＢＡディスク９とキャッシュ・メモリ６の間のデータの読み書きをＦＢＡコマンドを使用して実行するのは、データ・パス・サーバ（ＤＰＳ）５５であるマネージャは、全体の制御とキャッシュ・メモリ６の管理を行なう。

【００３４】ＣＫＤ－ＦＢＡ変換を行う際には三つのポイントが重要となる。すなわち、

ｉ．ＣＫＤのトラック・フォーマットのどこを残し、どこを削除するか？

ｉｉ．そのようにトラック・フォーマットを加工した後、どのようにして、元のレコードの位置を見つけるか？

ｉｉｉ．ＣＫＤのトラック容量をどのようにチェックするか、すなわち、トラック・オーバランの検知をどのように行うか？

が三つのポイントとして検討されなければならない。上記の三つのポイントについて考えてみると、他のバリエーションが考えられることが分かる。以下、順番に各ポイントについてバリエーションを考える。

【００３５】まず、ｉ．については、ＥＣＣやパディング等、元々ディスク装置側で付加していた情報をＤＫＣで生成してからＦＢＡに変換するのは無駄である。従って、「どこを残し、どこを削除するか？」ということは、事実上「どのギャップを残し、どのギャップを削除するのか？」ということになる（ＥＣＣやパディング以外の、ディフェクト情報のようなＤＫＣで管理していた制御情報や物理パラメータはギャップに含めて考えれば良い）。このギャップの残し方については、従来の方式も含め以下のようなバリエーションが考えられる（図６参照）。

＜ケース　ｉ－１＞　・・・「ｉＢＭ４３２１／４３３１プロセッサ互換性機能」の方法

・レコード間ギャップも含めすべてのギャップを削除してしまう。

＜ケース　ｉ－２＞　・・・従来の方法

・レコード間ギャップだけ残し、フィールド間ギャップは取り除く。

＜ケース　ｉ－３＞

・レコード間ギャップだけでなく、すべてのフィールド間ギャップも残す（ＥＣＣやパディングなどは取り除く）。

【００３６】これらの各方法は、それぞれ、ギャップ以

＜ケース　ＩＩ－１＞

・レコードの位置を示す特別の情報はいっさい持たない。

・レコードを探すときは，常にトラックの先頭のレコードから順番に読んでいき，各レコードのカウント・フィールドに含まれるＫＬ，ＤＬの値から次のレコードのカウント・フィールドの位置を計算する。

＜ケース　ＩＩ－２＞　・・・従来の方法

・部分的に、レコードの位置を示す情報をレコードそのものとは別に保持する。（従来の方法では、各ＦＢＡブロックの先頭に位置するレコードのカウント・フィールドの位置を保持する）

・位置を示されているレコードより後ろのレコードは＜ケースＩＩ－１＞と同じ様に、位置の分かっているレコードから順番に読みだし，各レコードのＫＬ，ＤＬから次のレコードの位置を計算する。

＜ケース　ＩＩ－３＞

・すべてのレコードの位置を示す情報をレコードそのものとは別に保持する。

・すべてのレコードはこの情報から位置を探す。

・レコードの位置情報は全レコード分を一括して一箇所に保持する方法（例えば，トラックの先頭）と，分散して保持する方法（例えば，各ＦＢＡブロックの先頭にそのＦＢＡブロックに含まれるレコードの位置を示す情報を保持する）が考えられる。

＜ケース　ＩＩ－４＞

・ＦＢＡ変換後の論理的なトラック・イメージ上でもなんらかのアドレス・マーク（ＡＭ）を生成し，これを目印にレコードを探す（メモリ上でもＡＭサーチを行う）。しかし，この方法はうまいアドレス・マークの生成法を具体的に考えついていないこと，メモリ上でＡＭサーチを行うために全データを順番に読み出すことはメモリ負荷を不必要に高め非効率的であることなどから，検討の範囲から除外する。

【００３７】次いで，ＩＩＩのトラック容量のチェック方法については，以下のような方法が考えられる（図８参照）。これらの方法は，Ｉのギャップの残し方によっては，適用できるものとできないものがある。

＜ケース　ＩＩＩ－１＞

・トラックの先頭からのすべてのレコードのＫＬ，ＤＬを知り、あいだのギャップやパディングの分（これらはすべて一意的にそのサイズが決っている）も全部足し込んで，既に消費されているＣＫＤフォーマット上でのトラック容量を計算する。

・従って、各レコードの位置が分かれば、そのレコードのメモリ上のアドレスから直ちに既に消費しているＣＫＤフォーマット上でのトラック容量が分かる。

＜ケース　ＩＩＩ－３＞

・各レコードにＣＫＤフォーマットでのトラックの先頭からの相対位置を示す情報（例えばセグメント・ナンバ）を残しておく。

・従って、各レコードの位置が分かれば、そのレコードに記録されている相対位置を示す情報から直ちに既に消費しているＣＫＤフォーマット上でのトラック容量が分かる。

【００３８】図８では＜ケース　ＩＩＩ－１＞と＜ケース　Ｉｉｉ－３＞は，すべてのギャップを取り除いているケース＜ケース　Ｉ－１＞で表現しているが，これらの方法はいずれもギャップがどのように残っていても構わない。また，＜ケース　Ｉｉｉ－２＞は，レコード間ギャップだけ残しているケース＜ケース　Ｉ－２＞で表現しているが，すべてのギャップを残すケース＜ケースＩ－３＞でも構わない。ただし，＜ケース　ＩＩＩ－２＞は，すべてのギャップを取り去る＜ケース　Ｉ－１＞では，ギャップ長の調節により，メモリ・アドレスを元のレコードの相対位置のまま維持することができないので，原理上，実現不可能である。これらのＩ，ＩＩ，ＩＩＩの各ケースの組み合せにより，（上述のように不可能な組み合わせ，また、無意味な組み合わせも有り）いくつかのＣＫＤ－ＦＢＡ変換方式ができることになる。従来の方式は＜ケース　Ｉ－２＞，＜ケース　ＩＩ－２＞，＜ケース　Ｉｉｉ－２＞の組み合せによるＣＫＤ－ＦＢＡ変換方式ということができる。

【００３９】図９に前述の各ケースの組み合せのうち有効なものの一覧を示す。各ケースの組み合せのうち、実現不可能なもの，無意味なものはあらかじめ除いてある。比較項目の欄の説明については，次節以降に示す。実現不可能なものとしては，前述した下記の組み合せがある。

●＜ケース　Ｉ－１＞と＜ケース　ＩＩＩ－２＞の組み合わせ

＜ケース　Ｉ－１＞はすべてのギャップを取り除く方式であるので、ギャップ長の調節により，メモリ・アドレスを元のレコードの相対位置のまま維持する＜ケースＩＩＩ－２＞のトラック容量のチェック方法は不可能である。また、下記の組み合わせは不可能ではないが、以下の理由により方式上無意味である。

ードを探すのにトラック上の最初のレコードから順番に読み出す必要がないという利点がある。しかし、＜ケース　ＩＩＩ－１＞はトラック容量をチェックするために、トラック上の全カウント・フィールドのデータが必要である。従って、これらの組み合せでは、結局、一度はトラック上の最初のレコードから順番に読み出すことになり、＜ケース　ＩＩ－２＞または＜ケース　ＩＩ－３＞の方式を採用した意味がなくなる。すなわち、＜ケース　ＩＩＩ－１＞はレコードの位置決め方式としてトラック上の全レコードを読み出す＜ケース　ＩＩ－１＞を採用したときだけ意味を持つトラック容量チェック方式ということができる。これら以外の組み合わせは、すべて図９にのせてある。図９の見方は、一行が一つの組み合せで実現されるＣＫＤ-ＦＢＡ変換方式を示しており、○のついている方式がその組み合せで採用されている方式である。従って、例えば一行目の方式は、「ギャップの残し方」の方式としては＜ケース　Ｉ－１＞を採用し、「レコード位置決め方法」については＜ケース　ＩＩ－１＞を、「トラック容量のチェック方法」としては＜ケース　ＩＩＩ－１＞を採用したＣＫＤ－ＦＢＡ変換方式ということになる。今後、このような方式を便宜上＜１，１，１＞と表記することにする。この表記法では、従来の方式は＜２，２，２＞と表現できる。

【００４０】以後、各ＣＫＤ－ＦＢＡ変換方式の比較を行うが、まず、そのために、各変換方式の比較ポイントを下記のように考える。

ａ．ＣＫＤフォーマットのトラック・イメージを磁気ディスク制御装置（ＤＫＣ）のキャッシュ・メモリ（以下、単にメモリともいう）上に展開する際の、メモリ容量の大小。・・・このメモリ容量は変換方式だけでなく、１トラック分の容量はそのときのトラック・フォーマット（レコード数の大小など）に大きく依存するため、しょせん最悪値にあわせて多めに取らざるを得ない。しかも、このメモリ・イメージをＦＢＡの物理ディスクに書くことになるのであるから、ＦＢＡディスクのブロック（セクタ）容量の整数倍となる。従って、少々の大小はあまり重大な問題ではない。主として、ギャップの残し方、レコードの位置決めのためなどに必要な付加情報の大小などによる。

ｂ．データ転送のやりやすさ・・・チャネルとの間で連続してデータ転送する部分などは、メモリ上でも連続していた方がやりやすい。例えば、キー・フィールドとデータ・フィールドは、メモリトでも連続していた方が転

単に手順の回数だけでなく、そのために必要なメモリ・アクセスの回数の多さなども考慮する必要がある。トラック・イメージは共通領域であるキャッシュ・メモリ上に展開するので、手順そのもの（例えば計算量）が少々多くても、メモリ・アクセス回数が少ない方がシステム全体では有利である。主として、このメモリ・アクセス回数はレコードの位置決め方法で決る。

ｄ．フォーマット・ライト時の手順・・・アップ・デート・ライト時は既にフォーマットされているレコードのＫＬ、ＤＬ見てライトすれば良いが、フォーマット・ライト時は、メモリ上で残トラック容量とホストから送られてきたレコードのＫＬ、ＤＬの整合性を調べる必要がある。すなわち、トラック・オーバランのチェックが必要である。さらに、方式によっては、レコード以外の制御情報をメンテナンスする必要があるので、そのための手順及び、メモリ・アクセスの多さなども考慮する必要がある。

以下、この比較ポイントをもとに各ＣＫＤ－ＦＢＡ変換方式の個々の特徴や、細かい分析を行う。

【００４１】＜１，１，１＞

・ギャップ：すべて削除

・レコード位置決め：トラックの最初のレコードから全レコード・リード

・トラック容量チェック：トラックの最初のレコードから全レコード・リード

ａ．メモリ容量

この方式は、ギャップもすべてなく、制御情報の付加もいっさいないので、メモリの利用効率は、もっとも良い。

ｂ．データ転送のやりやすさ

各フィールドがメモリ上で連続しているので、チャネルとの間で連続したフィールドのデータ転送する時、メモリ・アドレスはカウント・アップしていくだけでよい。

ｃ．目的レコードを見つけるときの手順

目的レコードを見つけるためには、トラックの先頭から該当レコードまでのレコード数だけメモリをアクセスする必要がある。すなわち、各レコードのカウント・フィールドのリードが必要になる。各レコードのカウント・フィールドを取得した後、それらのＫＬ，ＤＬから目的レコードの位置を計算していくのは、余分なギャップなどがないぶん、ＫＬ，ＤＬを足しこんでいけばよく単純である。

ｄ．フォーマット・ライト時の手順

・ギャップ：すべて削除
・レコード位置決め：ＦＢＡブロックに含まれる最初のレコードの位置を保持
・トラック容量チェック：各レコードにセグメント・ナンバ等の相対位置情報を保持
a．メモリ容量
ギャップはすべて削除するが、ＦＢＡのブロックに含まれる最初のレコードの位置や、各レコードのセグメント・ナンバ等を保持するための制御情報の分だけは、＜1，1，1＞より余分にメモリを使用する。
b．データ転送のやりやすさ
やはり、各フィールドがメモリ上で連続しているので、チャネルとの間で連続したフィールドのデータ転送する時、メモリ・アドレスはカウント・アップしていくだけでよい。しかし、単一フィールド内であっても、ＦＢＡのブロックにまたがったような場合は、メモリ上でもＦＢＡブロックに含まれる最初のレコードの位置情報などにより分断されるので、その位置情報の部分をスキップ・ディフェクトなどのように飛ばしてデータ転送を行う必要がある。
c．目的レコードを見つけるときの手順
目的レコードを探すためには、ＦＢＡのブロックの最初のレコードから目的レコードまでの間に存在するレコードの回数だけメモリをアクセスする必要がある。従って、＜1，1，1＞よりはメモリ・アクセス回数は少ない。ＦＢＡのブロックの最初のレコードの位置から目的レコードの位置を計算する方法はほとんど＜1，1，1＞と同じで単純である（カウント・フィールドのサイズとＫＬ，ＤＬを足し込んでいけば良い）。
d．フォーマット・ライト時の手順
トラック容量のチェックのためには、フォーマット・ライトしようとしているレコードの一つ前のレコードのセグメント・ナンバ（本来のＣＫＤフォーマットで、そのレコードのトラック上での相対位置を示す情報）とＫＬ、ＤＬから、後どれだけのサイズのレコードがライトできるか計算する。フォーマット・ライトしたときは、ライトしたレコード自身にそのレコードのセグメント・ナンバを書いておかなければならない。同時に、新たなＦＢＡブロック上に新たなカウント・フィールドが生じたときは、そのＦＢＡブロックに先頭のレコードのカウント・フィールドの位置を示す情報を書かなければならない。

【００４３】＜1，3，3＞
・ギャップ：すべて削除
・レコード位置決め：ＦＢＡブロックに含まれる最初のレコードの位置を保持
・トラック容量チェック：各レコードにセグメント・ナンバ等の相対位置情報を保持
a．メモリ容量
ギャップはすべて削除するが、ＦＢＡのブロックに含まれる最初のレコードの位置や、各レコードのセグメント・ナンバ等を保持するための制御情報の分だけは、＜1，1，1＞より余分にメモリを使用する。
b．データ転送のやりやすさ
やはり、各フィールドがメモリ上で連続しているので、チャネルとの間で連続したフィールドのデータ転送する時、メモリ・アドレスはカウント・アップしていくだけでよい。しかし、単一フィールド内であっても、ＦＢＡのブロックにまたがったような場合は、メモリ上でもＦＢＡブロックに含まれる最初のレコードの位置情報などにより分断されるので、その位置情報の部分をスキップ・ディフェクトなどのように飛ばしてデータ転送を行う必要がある。
c．目的レコードを見つけるときの手順
目的レコードを探すためには、ＦＢＡのブロックの最初のレコードから目的レコードまでの間に存在するレコードの回数だけメモリをアクセスする必要がある。従って、＜1，1，1＞よりはメモリ・アクセス回数は少ない。ＦＢＡのブロックの最初のレコードの位置から目的レコードの位置を計算する方法はほとんど＜1，1，1＞と同じで単純である（カウント・フィールドのサイズとＫＬ，ＤＬを足し込んでいけば良い）。
d．フォーマット・ライト時の手順
トラック容量のチェックのためには、フォーマット・ライトしようとしているレコードの一つ前のレコードのセグメント・ナンバ（本来のＣＫＤフォーマットで、そのレコードのトラック上での相対位置を示す情報）とＫＬ、ＤＬから、後どれだけのサイズのレコードがライトできるか計算する。フォーマット・ライトしたときは、ライトしたレコード自身にそのレコードのセグメント・ナンバを書いておかなければならない。同時に、必ずレコードの位置情報を保持している制御情報に、そのレコードのカウント・フィールドの位置を示す情報を書き足さなければならない。

【００４４】＜2，1，1＞
・ギャップ：フィールド間ギャップのみ削除、レコード間ギャップは残す
・レコード位置決め：トラックの最初のレコードから全レコード・リード
・トラック容量チェック：トラックの最初のレコードから全レコード・リード
a．メモリ容量
別に保持するため、＜1，2，3＞のようにＦＢＡのブロックに含まれる最初のレコードの位置だけ保持する場合より、さらにメモリ容量は多く必要である。各レコードにセグメント・ナンバなどを保持するのは＜1，2，3＞と同じである。
b．データ転送のやりやすさ
ほぼ、＜1，2，3＞と同じである。各フィールドがメモリ上で連続しているので、チャネルとの間で連続したフィールドのデータ転送する時、メモリ・アドレスはカウント・アップしていくだけでよい。レコードの位置情報をＦＢＡのブロックごとに分散して持つ場合は、単一フィールド内であっても、ＦＢＡのブロックにまたがる場合、メモリ上でもＦＢＡブロックに含まれるレコードの位置情報などにより分断されるので、その位置情報の部分をスキップ・ディフェクトなどのように飛ばしてデータ転送を行う必要がある。
c．目的レコードを見つけるときの手順
目的レコードを探すためには、レコードの位置情報を一回だけ読み取れば良いので、メモリ・アクセス回数は全方式の中で最小である。目的レコードの位置は、その情報からダイレクトに分かる。
d．フォーマット・ライト時の手順
＜1，2，3＞とトラック容量のチェック方法が同じであるので、フォーマット・ライト時の手順も＜1，2，3＞とほぼ同じである。トラック容量のチェックのためには、フォーマット・ライトしようとしているレコードの一つ前のレコードのセグメント・ナンバ（本来のＣＫＤフォーマットで、そのレコードのトラック上での相対位置を示す情報）とＫＬ、ＤＬから、後どれだけのサイズのレコードがライトできるか計算する。フォーマット・ライトしたときは、ライトしたレコード自身にそのレコードのセグメント・ナンバを書いておかなければならない。同時に、必ずレコードの位置情報を保持している制御情報に、そのレコードのカウント・フィールドの位置を示す情報を書き足さなければならない。

【００４４】＜2，1，1＞
・ギャップ：フィールド間ギャップのみ削除、レコード間ギャップは残す
・レコード位置決め：トラックの最初のレコードから全レコード・リード
・トラック容量チェック：トラックの最初のレコードから全レコード・リード
a．メモリ容量

＊，＊，２＞方式に比べメモリ容量は少なくできるが、逆に言うと、レコード間ギャップを残した意味もない。

b．データ転送のやりやすさ

レコード間ギャップは存在するが、フィールド間ギャップは無いので．チャネルとの間で連続したフィールドのデータを転送するとき（C→K→D）．メモリ・アドレスはカウント・アップするだけでよい。余分な制御情報もないので、単一フィールドがＦＢＡのブロックにまたがっても、メモリ上でそのフィールドが分断されることもない。

c．目的レコードを見つけるときの手順

＜１，１，１＞とほぼ同じである。目的レコードを見つけるためには．トラックの先頭から該当レコードまでのレコード数だけメモリをアクセスする必要がある。すなわち．各レコードのカウント・フィールドのリードが必要となる。各レコードのカウント・フィールドを取得した後．それらのＫＬ，ＤＬから目的レコードの位置を計算していくには．ＫＬ，ＤＬに加えレコード間ギャップを足しこんでいけばよく単純である。

d．フォーマット・ライト時の手順

これも＜１，１，１＞とほぼ同じである。トラック容量をチェックするためにも．トラックの先頭から該当レコードまでのレコード数だけメモリをアクセスする必要がある。トラック容量のチェックのためには、削除してしまったフィールド間ギャップなどの分も換算してやる必要がある。フォーマット・ライト時．更新しなければならない制御情報はない。

【００４５】＜２，２，２＞　・・・従来の方式

・ギャップ：フィールド間ギャップのみ削除、レコード間ギャップは残す

・レコード位置決め：ＦＢＡのブロックに含まれる最初のレコード位置を保持

・トラック容量チェック：ギャップ長の調節により各レコードの相対位置を維持

a．メモリ容量

フィールド間ギャップなどは削除するものの、その分レコード間ギャップを伸ばして各レコードの相対位置を元のＣＫＤフォーマットのまま維持するので、結局必要なメモリ容量は．元のＣＫＤフォーマットのトラック容量と同じで、さらにＦＢＡのブロックに含まれる最初のレコードの位置情報の分だけ余分にメモリが必要である。

b．データ転送のやりやすさ

レコード間ギャップは存在するが．フィールド間ギャップ・ディフェクトなどのように飛ばしてデータ転送を行う必要がある。

c．目的レコードを見つけるときの手順

目的レコードを探すためには、ＦＢＡのブロックの最初のレコードから目的レコードまでの間に存在するレコードの回数だけメモリをアクセスする必要がある。従って．＜２，１，１＞よりはメモリ・アクセス回数は少ない。ＦＢＡのブロックの最初のレコードの位置から目的レコードの位置を計算する方法はほとんど＜２，１，１＞と同じで単純である（カウント・フィールドのサイズとＫＬ，ＤＬさらにレコード間ギャップをどんどん足し込んでいけば良い）。また．各レコードのＣＫＤフォーマットでの相対位置をメモリ上でも維持しているので、Ｓｅｔ　Ｓｅｃｔｏｒの値から目的レコードの含まれるＦＢＡのブロックを正確に求めることができる。

d．フォーマット・ライト時の手順

トラック容量のチェックには、各レコードのＣＫＤフォーマットでの相対位置をメモリ上でも維持しているので．フォーマット・ライトしようとしているレコードのメモリ上のアドレスが分かれば（c．の手順で分かる）．直ちに後どれだけのサイズのレコードがライトできるか計算できる。フォーマット・ライトして．新たなＦＢＡブロック上に新たなカウント・フィールドが生じたときは、そのＦＢＡブロックに先頭のレコードのカウント・フィールドの位置を示す情報を書かなければならない。

【００４６】＜２，２，３＞

・ギャップ：フィールド間ギャップのみ削除、レコード間ギャップは残す

・レコード位置決め：ＦＢＡのブロックに含まれる最初のレコード位置を保持

・トラック容量チェック：各レコードにセグメント・ナンバ等の相対位置情報を保持

a．メモリ容量

フィールド間ギャップなどは削除し．レコード間ギャップを残すのは＜２，２，２＞と同じであるが、残したレコード間ギャップによりレコードの相対位置を維持しようということではないので．レコード間ギャップを無理に伸ばす必要はない（従って、やはりギャップを残す意味は薄くなる）。しかし．ＦＢＡのブロックに含まれる最初のレコードの位置情報の分は余分に必要である。また．各レコードに含まれるセグメント・ナンバの分も必要であるので　結局メモリ容量としては＜２．１．１＞

るセグメント・ナンバなどはカウント・フィールドに含まれるので、データ転送時C→Kの邪魔になることはない）。しかし、単一フィールド内であっても、FBAのブロックにまたがったような場合は、メモリ上でもFBAブロックに含まれる最初のレコードの位置情報などの制御情報により分断されるので、その制御情報の部分をスキップ・ディフェクトなどのように飛ばしてデータ転送を行う必要がある。

c．目的レコードを見つけるときの手順

これも＜2，2，2＞と全く同じである。目的レコードを探すためには、FBAのブロックの最初のレコードから目的レコードまでの間に存在するレコードの回数だけメモリをアクセスする必要がある。従って、＜1，1，1＞よりはメモリ・アクセス回数は少ない。FBAのブロックの最初のレコードの位置から目的レコードの位置を計算する方法はほとんど＜1，1，1＞と同じで単純である（カウント・フィールドのサイズとKL、DLさらにレコード間ギャップをどんどん足し込んでいけば良い）。

d．フォーマット・ライト時の手順

トラック容量のチェックのためには、フォーマット・ライトしようとしているレコードの一つ前のレコードのセグメント・ナンバ（本来のCKDフォーマットで、そのレコードのトラック上での相対位置を示す情報）とKL、DLから、後どれだけのサイズのレコードがライトできるか計算する。フォーマット・ライトしたときは、ライトしたレコード自身にそのレコードのセグメント・ナンバを計算して書いておかなければならない。同時に、新たなFBAブロック上に新たなカウント・フィールドが生じたときは、そのFBAブロックに先頭のレコードのカウント・フィールドの位置を示す情報を書かなければならない。

【0047】＜2，3，2＞

・ギャップ：フィールド間ギャップのみ削除、レコード間ギャップは残す

・レコード位置決め：すべてのレコードの相対位置情報を保持

・トラック容量チェック：ギャップ長の調節により各レコードの相対位置を維持

a．メモリ容量

残したレコード間ギャップ長の調節により各レコードの相対位置を元のCKDフォーマットのまま維持するのであるから、＜2，2，2＞と同じ様に　必要なメモリ容

＜2，2，2＞、＜2，2，3＞と全く同じである。レコード間ギャップは存在するが、フィールド間ギャップは無いので、チャネルとの間で連続したフィールドのデータを転送するとき（C→K→D）、メモリ・アドレスはカウント・アップするだけでよい。しかし、レコードの位置情報をFBAのブロックごとに分散して持つ場合は、単一フィールド内であっても、FBAのブロックにまたがったような場合は、メモリ上でもFBAブロックに含まれるレコードの位置情報などにより分断されるので、その位置情報の部分をスキップ・ディフェクトなどのように飛ばしてデータ転送を行う必要がある。

c．目的レコードを見つけるときの手順

＜1，3，3＞と全く同じである。目的レコードを探すためには、レコードの位置情報を一回だけ読み取れば良いので、メモリ・アクセス回数は全方式の中で最小である。目的レコードの位置は、その情報からダイレクトに分かる。また、各レコードのCKDフォーマットでの相対位置をメモリ上でも維持しているので、Set Sectorの値から目的レコードの含まれるFBAのブロックを正確に求めることができる。

d．フォーマット・ライト時の手順

＜2，2，2＞と＜1，3，3＞の中間的な手順となる。トラック容量のチェックには、各レコードのCKDフォーマットでの相対位置をメモリ上でも維持しているので、フォーマット・ライトしようとしているレコードのメモリ上のアドレスが分かれば（c．の手順で分かる）、直ちに後どれだけのサイズのレコードがライトできるか計算できる。フォーマット・ライトしたときは、必ずレコードの位置情報を保持している制御情報に、そのレコードのカウント・フィールドの位置を示す情報を書き足さなければならない。

【0048】＜2，3，3＞

・ギャップ：フィールド間ギャップのみ削除、レコード間ギャップは残す

・レコード位置決め：すべてのレコードの相対位置情報を保持

・トラック容量チェック：各レコードにセグメント・ナンバ等の相対位置情報を保持

a．メモリ容量

フィールド間ギャップなどは削除し、レコード間ギャップを残すのは＜2，3，2＞と同じであるが、残したレコード間ギャップによりレコードの相対位置を維持しようということではないので　レコード間ギャップを無視

b．データ転送のやりやすさ
これは＜2，3，2＞と全く同じである。レコード間ギャップは存在するが、フィールド間ギャップは無いので．チャネルとの間で連続したフィールドのデータを転送するとき（C→K→D）．メモリ・アドレスはカウント・アップするだけでよい（各レコードに保持されているセグメント・ナンバなどはカウント・フィールドに含まれるので、データ転送時C→Kの邪魔になることはない）。また、レコードの位置情報をFBAのブロックごとに分散して持つ場合は．単一フィールド内であっても．FBAのブロックにまたがる場合．メモリ上でもFBAブロックに含まれるレコードの位置情報などにより分断されるので．その位置情報の部分をスキップ・ディフェクトなどのようにして飛ばしてデータ転送を行う必要がある。
c．目的レコードを見つけるときの手順
これも＜2，3，2＞と全く同じである。目的レコードを探すためには．レコードの位置情報を一回だけ読み取れば良いので．メモリ・アクセス回数は全方式の中で最小である。目的レコードの位置は、その情報からダイレクトに分かる。
d．フォーマット・ライト時の手順
トラック容量のチェックのためには．フォーマット・ライトしようとしているレコードの一つ前のレコードのセグメント・ナンバ（本来のCKDフォーマットで．そのレコードのトラック上での相対位置を示す情報）とKL．DLから．後どれだけのサイズのレコードがライトできるか計算する。フォーマット・ライトしたときは、ライトしたレコード自身にそのレコードのセグメント・ナンバを計算して書いておかなければならない。同時に．必ずレコードの位置情報を保持している制御情報に．そのレコードのカウント・フィールドの位置を示す情報を書き足さなければならない。
【0049】＜3，1，1＞
・ギャップ：レコード間ギャップ、フィールド間ギャップとも残す
・レコード位置決め：トラックの最初のレコードから全レコード・リード
・トラック容量チェック：トラックの最初のレコードから全レコード・リード
a．メモリ容量
余分な制御情報の付加はないが、レコード間ギャップ、フィールド間ギャップとも残すので　ほぼCKDフォーマット上でもレコード間．フィールド間ともにギャップが存在するので．連続したフィールドのデータ転送をチャネルとの間で行うとき．メモリ・アドレスは単純にカウント・アップするだけではなく、ギャップの分を飛ばして．再設定する必要がある。しかし．余分な制御情報はないので、単一フィールドがFBAのブロックにまたがっても、メモリ上でそのフィールドが分断されることはない。
c．目的レコードを見つけるときの手順
＜1，1，1＞とほぼ同じである。目的レコードを見つけるためには．トラックの先頭から該当レコードまでのレコード数だけメモリをアクセスする必要がある。すなわち．各レコードのカウント・フィールドのリードが必要になる。各レコードのカウント・フィールドを取得した後．それらのKL，DLから目的レコードの位置を計算していくには．KL，DLに加えレコード間ギャップ．フィールド間ギャップを足しこんでいけばよく単純である。
d．フォーマット・ライト時の手順
これも＜1，1，1＞とほぼ同じである。トラック容量をチェックするためにも．トラックの先頭から該当レコードまでのレコード数だけメモリをアクセスする必要がある。トラック容量のチェックのためには、ギャップはそのまま残っているが、ECCやパディング・データなどの分は換算してやる必要がある。ECCの分なども含めぴったり同じ容量をギャップ長で補正していれば、メモリ・アドレスから直ちに残トラック容量が分かるが、これは＜3，2，2＞方式となる。フォーマット・ライト時．更新しなければならない制御情報はない。
【0050】＜3，2，2＞
・ギャップ：レコード間ギャップ、フィールド間ギャップとも残す
・レコード位置決め：FBAのブロックに含まれる最初のレコードの位置を保持
・トラック容量チェック：ギャップ長の調節により各レコードの相対位置を維持
a．メモリ容量
レコード間ギャップ、フィールド間ギャップとも残っているが、ギャップ長を調節して各レコードの相対位置を元のCKDフォーマットのまま維持するので、結局必要なメモリ容量は＜2，2，2＞と同じで、元のCKDフォーマットのトラック容量と、FBAのブロックに含まれる最初のレコードの位置情報を加えた分だけ必要であ

さらに、＜3，1，1＞と異なり、単一フィールド内であっても、ＦＢＡのブロックにまたがったような場合は、メモリ上でもＦＢＡブロックに含まれる最初のレコードの位置情報などにより分断されるので、その位置情報の部分をスキップ・ディフェクトなどのように飛ばしてデータ転送を行う必要がある。

c．目的レコードを見つけるときの手順

＜2，2，2＞と全く同じである。目的レコードを探すためには、ＦＢＡのブロックの最初のレコードから目的レコードまでの間に存在するレコードの回数だけメモリをアクセスする必要がある。従って、＜3，1，1＞よりはメモリ・アクセス回数は少ない。ＦＢＡのブロックの最初のレコードの位置から目的レコードの位置を計算する方法はほとんど＜3，1，1＞と同じで単純である（カウント・フィールドのサイズとＫＬ，ＤＬさらにレコード間ギャップ、フィールド間ギャップをどんどん足し込んでいけば良い）。また、各レコードのＣＫＤフォーマットでの相対位置をメモリ上でも維持しているので、Ｓｅｔ　Ｓｅｃｔｏｒの値から目的レコードの含まれるＦＢＡのブロックを正確に求めることができる。

d．フォーマット・ライト時の手順

これも＜2，2，2＞と全く同じである。トラック容量のチェックには、各レコードのＣＫＤフォーマットでの相対位置をメモリ上でも維持しているので、フォーマット・ライトしようとしているレコードのメモリ上のアドレスが分かれば（c．の手順で分かる）、直ちに後どれだけのサイズのレコードがライトできるか計算できる。フォーマット・ライトして、新たなＦＢＡブロック上に新たなカウント・フィールドが生じたときは、そのＦＢＡブロックに先頭のレコードのカウント・フィールドの位置を示す情報を書かなければならない。

【００５１】＜3，2，3＞

・ギャップ：レコード間ギャップ、フィールド間ギャップとも残す

・レコード位置決め：ＦＢＡのブロックに含まれる最初のレコードの位置を保持

・トラック容量チェック：各レコードにセグメント・ナンバ等の相対位置情報を保持

a．メモリ容量

レコード間ギャップ、フィールド間ギャップとも残すので、ほぼＣＫＤフォーマットのトラック・イメージがそのまま残され、さらに、ＦＢＡのブロックに含まれる最初のレコードの位置情報の分も余分に必要であるためメ

もレコード間、フィールド間ともにギャップが存在するので、連続したフィールドのデータ転送をチャネルとの間で行うとき、メモリ・アドレスは単純にカウント・アップするだけではなく、＜3，1，1＞と同様、ギャップの分を飛ばして、再設定する必要がある。さらに、＜3，1，1＞と異なり、単一フィールド内であっても、ＦＢＡのブロックにまたがったような場合は、メモリ上でもＦＢＡブロックに含まれる最初のレコードの位置情報などにより分断されるので、その位置情報の部分をスキップ・ディフェクトなどのように飛ばしてデータ転送を行う必要がある。

c．目的レコードを見つけるときの手順

これも＜3，2，2＞と全く同じである。目的レコードを探すためには、ＦＢＡのブロックの最初のレコードから目的レコードまでの間に存在するレコードの回数だけメモリをアクセスする必要がある。従って、＜3，1，1＞よりはメモリ・アクセス回数は少ない。ＦＢＡのブロックの最初のレコードの位置から目的レコードの位置を計算する方法はほとんど＜3，1，1＞と同じで単純である（カウント・フィールドのサイズとＫＬ，ＤＬさらにレコード間ギャップ、フィールド間ギャップをどんどん足し込んでいけば良い）。

d．フォーマット・ライト時の手順

トラック容量のチェックのためには、フォーマット・ライトしようとしているレコードの一つ前のレコードのセグメント・ナンバ（本来のＣＫＤフォーマットで、そのレコードのトラック上での相対位置を示す情報）とＫＬ、ＤＬから、後どれだけのサイズのレコードがライトできるか計算する。フォーマット・ライトしたときは、ライトしたレコード自身にそのレコードのセグメント・ナンバを書いておかなければならない。同時に新たなＦＢＡブロック上に新たなカウント・フィールドが生じたときは、そのＦＢＡブロックに先頭のレコードのカウント・フィールドの位置を示す情報を書かなければならない。

【００５２】＜3，3，2＞

・ギャップ：レコード間ギャップ、フィールド間ギャップとも残す

・レコード位置決め：すべてのレコードの相対位置情報を保持

・トラック容量チェック：ギャップ長の調節により各レコードの相対位置を維持

a．メモリ容量

ドの分を保持するので、メモリ容量も＜３，２，２＞より多くなり、＜２，３，２＞と同様全方式中最大である。

b．データ転送のやりやすさ

これは、＜３，２，２＞、＜３，２，３＞と全く同じである。メモリ上でもレコード間、フィールド間ともにギャップが存在するので、連続したフィールドのデータ転送をチャネルとの間で行うとき、メモリ・アドレスは単純にカウント・アップするだけではなく、＜３，１，１＞と同様、ギャップの分を飛ばして、再設定する必要がある。さらに、＜３，１，１＞と異なり、レコードの位置情報をＦＢＡのブロックごとに分散して持つ場合は、単一フィールド内であっても、ＦＢＡのブロックにまたがったような場合は、メモリ上でもＦＢＡブロックに含まれるレコードの位置情報などにより分断されるので、その位置情報の部分をスキップ・ディフェクトなどのように飛ばしてデータ転送を行う必要がある。

c．目的レコードを見つけるときの手順

＜１，３，３＞、＜２，３，３＞などと全く同じである。目的レコードを探すためには、レコードの位置情報を一回だけ読み取れば良いので、メモリ・アクセス回数は全方式の中で最小である。目的レコードの位置は、その情報からダイレクトに分かる。また、各レコードのＣＫＤフォーマットでの相対位置をメモリ上でも維持しているので、Ｓｅｔ　Ｓｅｃｔｏｒの値から目的レコードの含まれるＦＢＡのブロックを正確に求めることができる。

d．フォーマット・ライト時の手順

＜２，３，２＞と全く同じである。トラック容量のチェックには、各レコードのＣＫＤフォーマットでの相対位置をメモリ上でも維持しているので、フォーマット・ライトしようとしているレコードのメモリ上のアドレスが分かれば（ｃ．の手順で分かる）、直ちに後どれだけのサイズのレコードがライトできるか計算できる。フォーマット・ライトしたときは、必ずレコードの位置情報を保持している制御情報に、そのレコードのカウント・フィールドの位置を示す情報を書き足さなければならない。

【００５３】＜３，３，３＞

・ギャップ：レコード間ギャップ、フィールド間ギャップとも残す

・レコード位置決め：すべてのレコードの相対位置情報を保持

よりもさらに多く、最大に近くなる。しかし、ＣＫＤフォーマットにおけるレコードの相対位置を積極的に維持しようと言うのではないので、ＥＣＣやパディングの分まではメモリ容量に加える必要はない。そのかわり、ギャップを残した意味はあまりなくなる。

b．データ転送のやりやすさ

これは、＜３，２，２＞、＜３，２，３＞，＜３，３，２＞と全く同じである。メモリ上でもレコード間、フィールド間ともにギャップが存在するので、連続したフィールドのデータ転送をチャネルとの間で行うとき、メモリ・アドレスは単純にカウント・アップするだけではなく、＜３，１，１＞と同様、ギャップの分を飛ばして、再設定する必要がある。さらに、＜３，１，１＞と異なり、レコードの位置情報をＦＢＡのブロックごとに分散して持つ場合は、単一フィールド内であっても、ＦＢＡのブロックにまたがったような場合は、メモリ上でもＦＢＡブロックに含まれるレコードの位置情報などにより分断されるので、その位置情報の部分をスキップ・ディフェクトなどのように飛ばしてデータ転送を行う必要がある。

c．目的レコードを見つけるときの手順

これは＜３，３，２＞と全く同じである。目的レコードを探すためには、レコードの位置情報を一回だけ読み取れば良いので、メモリ・アクセス回数は全方式の中で最小である。目的レコードの位置は、その情報からダイレクトに分かる。

d．フォーマット・ライト時の手順

＜２，３，３＞と全く同じである。トラック容量のチェックのためには、フォーマット・ライトしようとしているレコードの一つ前のレコードのセグメント・ナンバ（本来のＣＫＤフォーマットで、そのレコードのトラック上での相対位置を示す情報）とＫＬ、ＤＬから、後どれだけのサイズのレコードがライトできるか計算する。フォーマット・ライトしたときは、必ずライトしたレコード自身にそのレコードのセグメント・ナンバを書いておかなければならない。同時に、レコードの位置情報を保持している制御情報に、そのレコードのカウント・フィールドの位置を示す情報を書き足さなければならない。

【００５４】図９の右半分に前述した各方式の比較結果をまとめた。「比較項目」の欄には前述した４つの比較項目を記載している。各比較項目にはその重要度に応じ５段階のグレード付けをしている。５がもっとも重要な

の比較項目のグレードの値を掛けたものを全部の比較項目について合計している。従って、合計得点の高い変換方式ほど優れた変換方式ということになる。図9では、この合計得点が従来の方式より高い変換方式の合計得点欄に＊を付している。

【0055】図9の合計得点は多分に主観的な面もあり、あくまで目安である。しかし、試しに点数の与え方や各比較項目のグレードの値を多少振っても、従来の方式と同等以上の得点になる変換方式は、やはり図9と同じものであった。従って、今回の比較項目で考える限り、良い目安になると考えられる。従来の方式がトップにならなかった理由は、おもに目的レコードを見つけるときの手順におけるメモリ・アクセスの回数である。従来の方式においては、FBAのブロック中の最初のレコードは一発で見つけることができるが、その後ろのレコードはそこから順番に後続のレコードのカウント・フィールドを見つけて読んでいかなくてはならない。従来の方式がこの点を重視しなかった理由は下記の2点が考えられる。

（1）レコードの位置情報を持つ単位の問題

従来の方式はFBAディスクのブロック（≒セクタ）ごとにそこに含まれる最初のレコードの位置情報を持つとしている。現実のSCSIディスクなどでは、このブロックの大きさはせいぜい1～2KB程度であるのに対し、そこに記録するCKDのレコードのサイズは4KB程度がもっとも多い。従って、現実には一つのFBAブロックにたくさんのCKDレコードが含まれるケースよりも、図10のように一つのCKDレコードが複数のFBAブロックにまたがり、結局、一つのFBAブロックに含まれるCKDレコードは1個以下というケースの方が多いと考えられる。従って、Set Sectorにより、目的のレコードが含まれるFBAブロックさえきちんと分かればほとんど1回のメモリ・アクセスで目的レコードを見つけることができるので、この方式でもメモリ・アクセス回数が多くならない。

（2）トラック・バッファの置き場所の問題

従来の方式においては、このようなCKD－FBA変換を行うエミュレータは、チャネル・プロセッサとFBAディスクの間のどこにあってもよいとされているが（もちろんDKCでもよいことになる）、エミュレータはチャネル・プロセッサ自体が実行している。すなわち、ディスク・キャッシュをベースとした大規模なシステムはあまり意識していない。従って、トラック・イメージを

【0056】これらの点から従来はあまりメモリ・アクセスの回数という点を重視しなかったと考えられるが、上記の2点はそれぞれ以下のような問題をはらんでいる。

（1）レコードの位置情報を持つ単位の問題

従来の方式はFBAディスクのブロック（≒セクタ）ごとにそこに含まれる最初のレコードの位置情報を持つとしているが、非効率的である。図11に示すように、ディスク・キャッシュをベースにした4台のデータ・ディスクと1台の冗長ディスクをもつアレイ型ディスク状態（RAID）を考える場合、物理ディスクへのリード／ライトの単位はFBAのブロックの様な細かい単位ではなく、例えば、CKDの1トラック分（例えば48KB）を各データ・ディスクに分割した1／4トラック分である。従って、レコードの位置情報もFBAごとより、この1／4トラックごとに持った方がよい（図11参照）。1／4トラックごとに位置情報を持つのであれば、1トラック分の位置情報は4箇所に集約されるので、CKDレコードがメモリ上でこの位置情報により分断されるのも、この4箇所ですむ。しかし、従来方式のようにFBAのブロックごとにレコードの位置情報を持つとすると、1トラックの分断箇所も1トラックに含まれるFBAブロックの個数になる。（1ブロック2KBとすると、1トラックで約24箇所）すなわち、4KBのCKDレコードであれば必ず1レコード中1～2箇所分断されることになる。この分断箇所は特別なハードウェア（H／W）を用意しない限り磁気ディスク制御装置のファームウェア（F／W）で処理することになる。そのために数十μsecかかるとし、チャネルの転送レートを4．5MB／secとすれば、4KBのレコードを転送するのにかかる時間が889μsecであるから、1レコード当り数%～10%近い性能ダウンにつながる。これを避けるために図11のように、1／4トラックごとに位置情報を持たせることにすると、今度はその中に含まれるCKDレコード数が多くなるので、従来方式ではメモリ・アクセス回数が多くなってしまう。

【0057】（2）トラック・バッファの置き場所の問題

従来の方式のように、トラック・バッファがチャネル・プロセッサ自身のローカル・メモリにあり、そのチャネル・プロセッサがCKD－FBAエミュレーションを行うような場合は、そこへのメモリ・アクセス回数が少々多くても、あまり問題にはならない。ところが ディス

ばならないので．各チャネルとのデータ転送をつかさどるＤＫＣ内のチャネル・パス・サーバ（ＣＰＳ）がエミュレーションを行うことになる。従って、従来の方式では目的レコードを見つけるために、各チャネル・パス・サーバは共有部分であるキャッシュ・メモリ上のトラック・バッファを頻繁にアクセスすることになる。このキャッシュ・メモリへのアクセスはチャネル・パス・サーバだけでなく．デバイス側からも行われる。つまりこのキャッシュ・メモリはリード／ライトが高速で行われるシステムのボトル・ネック部分である。このような部分に細かい単位で頻繁にアクセスすることは、それだけで著しく内部バスのデータ転送能力を低くする原因となる。従って、このキャッシュ・メモリへのアクセスは少々データ量が多くなっても．１回でアクセスする方式の方がよい。以上のような理由から、メモリ・アクセスの回数という点にも注目しているため．その部分で従来の方式は若干点数を下げ、トップになっておらず．従来の方式より高得点をあげているのは、このメモリ・アクセス回数の点で有利な＜ケース　ｉｉ－３＞を採用している変換方式の中のものである。

【００５８】次に「ギャップの削除方法とメモリ容量の評価」について説明する。ここでは．ＣＫＤフォーマットのトラック・イメージをキャッシュ・メモリ（トラック・バッファ）上に展開してＦＢＡディスクに記録する際．ギャップも含めてメモリ上に展開するか否かについて．前述した３方式についてそれぞれ具体的にメモリ容量がどの程度異なるか評価する。

【００５９】ＣＫＤフォーマットのトラック・イメージをキャッシュ・メモリ（トラック・バッファ）上に展開してＦＢＡディスクに記録する際、ギャップも含めてメモリ上に展開するか否かについては．

＜ケース　ｉ－１＞

・レコード間ギャップも含めすべてのギャップを削除してしまう。

＜ケース　ｉ－２＞

・レコード間ギャップだけ残し、フィールド間ギャップは取り除く。

＜ケース　ｉ－３＞

・レコード間ギャップだけでなく、すべてのフィールド間ギャップも残す（ＥＣＣやパディングなどは取り除く）。

という３種類の方式が考えられる。ここでは、これらの各方式ごとにＣＫＤフォーマット１トラック分のトラック

のＣＫＤトラックのまま維持する方式）と組み合わせた場合は、トラック・イメージのメモリ容量は常に元のＣＫＤフォーマットと等しくなる。従って、ここではギャップ長の調整は行なわない場合について見積りを行う。

【００６０】ＣＫＤフォーマットでは．ギャップの数は１トラック中のレコード数及びキー・フィールドの有無などに依存するため、ギャップの有無によるメモリ容量の大小を見積もるには、見積もりを行うトラック中のレコード数などのトラック・フォーマットを規定する必要がある。ここでは極端なケースとして．レコード数がもっとも多い場合．もっとも少ない場合．また、標準的なケース等、図１２の４ケースについて見積もりを行う。いずれも、ＣＫＤフォーマットとしては図３３を想定している。それぞれのケースにおけるトラック当りのレコード数の計算は図１３、図１４、図１５、図１６に示してある。図１２のフォーマット２，３は、ここでは述べないが「ディスク・アクセスのワークロード・タイプ調査」という調査に基づいて定めた。また、ジャーナル・ファイルのレコード・サイズは２００Ｂ～３２ＫＢであるが．もっともよく用いられているサイズは２００Ｂ～３００Ｂくらいであるので．図１２では２００Ｂを用いている。

【００６１】メモリ容量を見積もるに当り、ここでは、下記のような条件で見積りを行う。

（１）トラック・フォーマットとしては図３３を使用する。

（２）各フィールドのＥＣＣ、スペース（Ｘ’ＦＦ’）、３２Ｂ境界にあわせるための０パディングは削除する。

（３）ＨＡ、カウント・フィールドのその他のパラメータはすべて残す。

（４）レコードの位置情報などのＣＫＤ－ＦＢＡ変換のための制御情報の容量はここでは見積もらない。これは．ギャップの削除方法とはまた別の条件であるので、別途見積もる。

結局．メモリ上ではギャップ以外の各フィールドの中身は図３３のＸで示した部分のようになるとして見積りを行う。見積りケースごとのトラック当りのレコード数を計算すると図１３～図１６のようになる。

【００６２】また、各ギャップ削除方法ごとのメモリ容量見積りは以下のようになる。

＜ケース　ｉ－１＞

・レコード間ギャップも含めすべてのギャップを削除し

35

よって、1トラック分の容量は、
HA+R0+Rn×93=28+29+29×93=2754
●フォーマット2（ジャーナル・ファイルの典型）
HAのサイズ　=40－12=28
R0のサイズ　=R0C+R0D=（40－12）+8=36
Rnのサイズ　=RnC+RnD=（40－12）+200=228
よって、1トラック分の容量は、
HA+R0+Rn×68=28+36+228×68=15568
●フォーマット3（ページング、スワッピング、VSAM）
HAのサイズ　=40－12=28
R0のサイズ　=R0C+R0D=（40－12）+8=36
Rnのサイズ　=RnC+RnD=（40－12）+4096=4124
よって、1トラック分の容量は、
HA+R0+Rn×10=28+36+4124×10=41304
●フォーマット4（最少レコード数のケース））
HAのサイズ　=40－12=28
R0のサイズ　=R0C+R0D=（40－12）+47988=48016
よって、1トラック分の容量は、
HA+R0=28+48016=48044
【0063】<ケース　I－2>
・レコード間ギャップだけ残し、フィールド間ギャップは取り除く（ECCやパディングなどは取り除く）。ただし、フィールド間ギャップやECC、スペース（X'FF'）、パディングなどを削除してもその分をギャップ長を伸ばしてつじつまを合わせることにすると、メモリ容量は元のCKDフォーマットと全く同じになる。従って、ここでは削除分のつじつま合わせは行わないケースで見積りを行う。
●フォーマット1（最多レコード数のケース）
HAのサイズ　=G1+HA=504+（40－12）=532
R0のサイズ　=G2C+R0C+R0D=248+（40－12）+1=277
Rnのサイズ　=G3+RnC+RnD=216+（4

36

R0のサイズ　=G2C+R0C+R0D=248+（40－12）+8=284
Rnのサイズ　=G3+RnC+RnD=216+（40－12）+200=444
よって、1トラック分の容量は、
HA+R0+Rn×68=532+284+444×68=31008
●フォーマット3（ページング、スワッピング、VSAM）
HAのサイズ　=G1+HA=504+（40－12）=532
R0のサイズ　=G2C+R0C+R0D=248+（40－12）+8=284
Rnのサイズ　=G3+RnC+RnD=216+（40－12）+4096=4340
よって、1トラック分の容量は、
HA+R0+Rn×10=532+284+4340×10=44216
●フォーマット4（最少レコード数のケース）
HAのサイズ　=G1+HA=504+（40－12）=532
R0のサイズ　=G2C+R0C+R0D=248+（40－12）+47988=48264
よって、1トラック分の容量は、
HA+R0=532+48264=48796
【0064】<ケース　I－3>
・レコード間ギャップだけでなく、すべてのフィールド間ギャップも残す（ECCやパディングなどは取り除く）。この場合も、ECCやスペース（X'FF'）、パディングなどを削除してもその分をギャップ長を伸ばしてつじつまを合わせることにすると、メモリ容量は元のCKDフォーマットと全く同じになる。従って、ここでは削除分のつじつま合わせは行わないケースで見積りを行う。
●フォーマット1（最多レコード数のケース）
HAのサイズ　=G1+HA=504+（40－12）=532
R0のサイズ　=G2C+R0C+G2+R0D=248+（40－12）+224+1=501
Rnのサイズ　=G3+RnC+G2+RnD=216+（40－12）+224+1=469
よって、1トラック分の容量は、
HA+R0+Rn×93=532+501+469×9

箇所に保持する方法について見積もらない理由は以下のとおりである。位置情報を一箇所で一括して保持していると、図19のようなアレイ型ディスク装置を考えた場合、4台のデータ・ディスクから1CKDトラック分のデータを読んでくるとき、目的レコードが記録されているディスクからのデータを読み終えても、位置情報が記録されているディスクからのデータを読み終らないと、目的レコードの記録位置を見つけることができず、ホストとのデータ転送を開始できない。従って、回転同期をサポートできなかった場合や、サポートしていてもスタンバイ・ディスクが加わって同期が崩れているときなどに平均回転待ち時間の増加につながる。

【0068】ここでは、見積り条件を以下のようなものとする。まず、レコード1個当りの保持情報として、レコード1個につき、下図のような4バイトの情報を保持するものとする。

レコードNo.：記録されているCKDレコードのレコードNo.である。<ケース ii－2>では制御上必ずしも必須ではないが、特に<ケース ii－3>では制御情報中にこのレコードNo.を持っていると、いちいち実際のレコードのカウント・フィールドを読まなくても一発で目的レコードを発見することができる。

セクタ値 ：CKDレコードとして記憶されている場合のセクタ値を記憶する。このセクタ値によりセットセクタコマンドで指定されたセクタ値との比較が可能になる。また、このセクタ値によりそのレコードのトラック先頭からの相対位置を求めることができ、既に消費した容量が判定できる。セクタは7つのセグメントから構成されているため、このセクタ値を用いてセグメントナンバを計算することが可能である。すなわち、セグメントナンバ＝セクタ値×7である。

メモリ・アドレス：記録されているCKDレコードのメモリ上でのバイト・アドレスである。2Bあれば64KBまでアドレッシングできるので、図33のフォーマットでは十分である。アドレスの起点は、CKDトラックの先頭にするか、位置情報の管理単位である各FBAブロック等の先頭にするかは、とりあえず問わない。

【0069】また、CKDレコードの位置情報を管理する単位（レコードの位置情報を保持する単位）としては図21の2ケースについて見積りを行う。上記の2つのケースにおいて、<ケース ii－2>と<ケース ii－3>のそれぞれについて記録されているCKDレコードの位置情報を持つためのメモリ容量を見積もる。見積り結果は以下のとおりである。

<ケース ii－2> ：先頭のレコードの位置情報だけ保持する。この方式では、位置情報の管理単位のサイズにかかわらず、常に管理単位ごとレコード1個分の位置情報しか保持しないので、位置情報のためのメモリ容量は管理単位ごとに4Bである。

<ケース ii－3> ：すべてのレコードの位置情報を保持する。この方式では、管理単位の大きさによって、そこに記録され得るCKDのレコード数が異なってくる。さらに、あるサイズの管理単位にどのくらいのCKDレコードが記録できるかは、ギャップの削除方法にも依存するが、ここでは、もっともレコード数が多くなる<ケース i－1>（すべてのギャップを削除する）を採用して見積りを行う。このときの1レコード当りの最少メモリ容量は、前述した「ギャップの削除方法とメモリ容量の評価」の「<ケース i－1>のフォーマット1（最多レコードのケース）」のRnのサイズより、29Bであることが分かっている。従って、各管理単位のサイズごとに記録できる最多レコード数NRは、位置情報のためのメモリ容量をαとすると

$$NR = \{(\text{管理単位のサイズ}) - \alpha\} \div 29B \quad \cdots (1)$$

※ると図22のようになる。

また、管理単位当りの位置情報のメモリ容量αはそのレコード数NRに4Bを掛ければよいので、

$$\alpha = NR \times 4B \quad \cdots (2)$$

で求められる。この2式を解くと、

$$NR = (\text{管理単位のサイズ}) / 33 \text{（小数点以下切捨て）} \quad \cdots (3)$$

となる。従って、（3）式で求めたNRを（2）式に代入することにより、ただちにαも求められる。ただし、図33のトラック・フォーマットでは1トラック内のレ

【0070】前述した「ギャップの削除方法とメモリ容量の評価」によれば、CKDレコードを記録するために正味必要なメモリ容量の最大値は、1トラック1レコード（R0のみ）でそのデータ・フィールドが最長の場合である。このとき必要なメモリ容量は、図17からHA、R0合わせて図23のようになる。CKDトラック1本のために用意するFBAディスクの容量を48KB（12KB×4）とすると、制御情報のために使用でき

＋（４０－１２）＋２２４＋２００＝６６８
よって、１トラック分の容量は、
ＨＡ＋Ｒ０＋Ｒｎ×６８＝５３２＋５０８＋６６８×６８＝４６４６４
●フォーマット３（ページング、スワッピング、ＶＳＡＭ）
ＨＡのサイズ　＝Ｇ１＋ＨＡ＝５０４＋（４０－１２）＝５３２
Ｒ０のサイズ　＝Ｇ２Ｃ＋Ｒ０Ｃ＋Ｇ２＋Ｒ０Ｄ＝２４８＋（４０－１２）＋２２４＋８＝５０８
Ｒｎのサイズ　＝Ｇ３＋ＲｎＣ＋Ｇ２＋ＲｎＤ＝２１６＋（４０－１２）＋２２４＋４０９６＝４５６４
よって、１トラック分の容量は、
ＨＡ＋Ｒ０＋Ｒｎ×１０＝５３２＋５０８＋４５６４×１０＝４６６８０
●フォーマット４（最少レコード数のケース））
ＨＡのサイズ　＝Ｇ１＋ＨＡ＝５０４＋（４０－１２）＝５３２
Ｒ０のサイズ　＝Ｇ２Ｃ＋Ｒ０Ｃ＋Ｇ２＋Ｒ０Ｄ＝２４８＋（４０－１２）＋２２４＋４７９８８＝４８４８８
よって、１トラック分の容量は、
ＨＡ＋Ｒ０＝５３２＋４８４８８＝４９０２０
【００６５】図１７に＜ケース　Ｉ－１＞～＜ケース　Ｉ－３＞までの３種類のギャップ削除方法について、それぞれフォーマット１～４までの４つのトラック・フォーマットをメモリ上に展開したときのメモリ容量を示す。図１８はこれをグラフで表したものである。
【００６６】図１７、図１８から分かるようにどのトラック・フォーマットでも、＜ケースＩ－１＞～＜ケース　Ｉ－３＞と残すギャップが多いほどメモリ容量も多く必要となっている。＜ケース　Ｉ－３＞ではすべてのギャップを残すため、どのフォーマットにおいてもほぼ元のＣＫＤフォーマットでの容量と等しいメモリ容量が必要となる。フィールド間ギャップを削除している＜ケース　Ｉ－２＞でも、トラック容量のチェック方式として、削除したフィールド間ギャップやＥＣＣ、パディングの分だけレコード間ギャップを伸ばし、各レコードのトラックの先頭からの相対位置を元のＣＫＤトラックのまま維持する方式をとると、必要なメモリ容量は＜ケース　Ｉ－３＞とほぼ同じになる。これに対し、すべてのギャップを削除する＜ケース　Ｉ－１＞では、特にレコード・サイズが小さいフォーマット１、２でメモリ容量が顕著に小さくなっている。９３レコードのフォーマット

ク分・・・１２ＫＢ）にすれば、余ったキャッシュ・メモリの領域には他のトラックのデータを書くことができる。従って、同一のキャッシュ・メモリ容量でより多くのトラックのデータを保持することができ、キャッシュ・メモリの利用率向上、引いてはヒット率の向上を期待することができる。
（２）１トラック分のイメージのデータ量が少なくてすむということは、物理ディスクへそのデータを書くときも少ないデータ量ですむということである。例えば、１トラック分のイメージのデータがＣＫＤフォーマットの１／２ですめば、物理ディスクへのデータ転送はデータ・ディスク１と２、及びパリティ・ディスクだけですみ、フォーマット・ライトでなければデータ・ディスク３と４にはデータを転送しなくてよい。これは、サブシステム内の各バス等の負荷軽減につながる。ただし、メモリ容量が少なくすんでも、物理ディスク上の領域は最大値に合わせ確保しておかないと制御が非常に複雑になる。メモリ上のトラック・イメージがＣＫＤフォーマットの半分だからといって、そのデータをディスクに書くときも空いた領域に他のトラックのデータを書いてしまうと、次にそのトラックが再フォーマットされて、トラック・イメージの量が増えてしまったとき、ディスク上の連続した領域にはトラック・イメージをライトできなくなり、ディスク上でどこか空いている領域を探してライトするようなことになる。トラック・イメージのデータ量が減った場合、その分物理ディスクの空き領域も有効に活用できればよりよいが、（１）、（２）のメリットだけでも十分魅力的である。
以上のように、メモリ容量の大小を実際に見積もった結果、ギャップをすべて残した場合と、ギャップをすべて削除した場合では、レコード数によっては必要なメモリ容量が数倍以上の開きがでることがわかる。
【００６７】次に、「レコードの位置情報のメモリ容量見積り」について説明する。目的レコードの位置決めを行う方式として、＜ケース　ＩＩ－１＞と＜ケースＩＩ－２＞と＜ケース　ＩＩ－３＞の３方式をあげた。これらのうち、＜ケース　ＩＩ－１＞を除く他の２方式は、いずれもレコードそのものとは別個に、メモリ上でのレコードの位置を示す制御情報を保持する方式である。ここでは、この位置決めのための制御情報に各方式でどの程度メモリ容量（＝ＦＢＡディスク上での容量）が必要なのか見積りを行う。ここでは＜ケース　ＩＩ－１＞を除く　＜ケース　ＩＩ－２＞と＜ケース　ＩＩ－３＞に

２サイズ１２ＫＢ）、位置情報のためのメモリ容量が少ない場合（１５０４Ｂ）と、レコードのためのメモリ容量がもっとも少なくて済む＜ケース　Ｉ－１＞で比べてみても位置情報のために使用できるメモリ容量約４００Ｂが不足してしまう。

【００７１】＜ケース　ＩＩ－３＞は目的レコードを探すためのメモリ・アクセス回数が少なく、従来の方式（＜ケース　ＩＩ－２＞）より有望であるので、この不足分に対する対策を考える必要がある。この対策としては、下記のようなものが考えられる。

対策１　物理ディスク１台当りにリザーブする容量を増やす。

ｅｘ．）　１２ＫＢ→１３ＫＢとすると、ＣＫＤトラック１本のために使用できる容量は、５２ＫＢとなり、不足は一挙に解決される。・・・・物理ディスクは５２ＫＢ／ＣＫＤトラック必要となり、４８ＫＢ／ＣＫＤトラックに比べてディスクエリアが無駄になるが、物理ディスクのコストが減少してきているので有効な方法のひとつである。

対策２　レコード１個当りに必要な制御情報の量を減らす。

ｅｘ．）　４Ｂ／レコード→２Ｂ／レコードとすると、制御情報に必要な総メモリ容量は、７５２Ｂとなり不足しなくなる。

対策３　制御情報の個数を減らす。

ｅｘ．）　この見積りでは、管理単位２の場合（サイズ１２ＫＢ）、４つの各管理単位ごとに最大９４個のレコードの位置情報を保持するように考えている。しかし、実際は４つの管理単位の合計でＭａｘ．９４個が必要であるので、うまく工夫して、制御情報の総容量を減らすことを考える。

【００７２】以上のように「レコードの位置情報のメモリ容量見積り」の結果は以下のとおりである。＜ケース　ＩＩ－２＞では、メモリ容量は不足せず問題ない。＜ケース　ＩＩ－３＞では条件によっても異なるが、位置情報のために使用できるメモリ容量が若干不足してしまう。＜ケース　ＩＩ－３＞は目的レコードを探すためのメモリ・アクセス回数が少なく、有望であるので、今後、この不足分に対する対策を考える必要がある。

【００７３】次に、「レコードの位置情報の制御方式の改善」について説明する。先の見積りの制御方式の問題点は以下のとおりである。前述した「レコードの位置情報のメモリ容量見積り」では、ＣＫＤレコードの位置情

４Ｂ×９４＝３７６Ｂ

だけの位置情報領域を見積もっていた。ＣＫＤトラック１本分のトラック・イメージはこの１２ＫＢの管理単位４個からなっており、データはこの１２ＫＢずつ４台のデータ・ディスクに分散して記録されるので、ＣＫＤトラック１本当りの位置情報のメモリ容量は、

３７６Ｂ×４＝１５０４Ｂ

となってしまう。しかし、これでは、ＣＫＤトラック１本当り９４×４＝３７６レコード分の位置情報の領域を確保していることになる。図３３のフォーマットでは、１ＣＫＤトラックの最大レコード数は９４レコード（Ｒ０含む）であるので、１２ＫＢの各管理単位に記録され得る最大レコード数がそれぞれ９４個であっても、同時に全部の管理単位に９４個ずつのレコードが記録されることはあり得ない。従って、９４×３＝２８２レコード分の位置情報の領域は無駄になっていることになる。

【００７４】このような問題点は、レコードの管理単位内の位置情報領域のサイズを固定にしようとしたことから発生している。位置情報領域のサイズを固定にすると、（１）管理単位にまたがって記録されているレコードのデータ転送を行うときも、飛び越さなければならない位置情報領域のサイズが常に一定なので、制御が簡単であること、（２）フォーマット・ライト時も、ＣＫＤレコードを書き込めるメモリの領域が固定なので、制御が簡単であること、などの利点が発生する。しかし、位置情報領域のサイズを固定にする限り、そのサイズは最大値に合わせざるを得ず、管理単位ごとに９４レコード分の領域を確保しておくほかない。これに対し、位置情報領域のサイズを可変にすると、実際に記録されているレコードの分だけしか位置情報領域を使用しないような制御も可能であるが、上記のような利点は基本的に失われてしまう。位置情報の総容量を減らすためには、各管理単位の位置情報領域のサイズを可変にすることが必須であるが、このような欠点をうまく回避することが重要である。

【００７５】図２５に位置情報領域のサイズを可変にした改善案を示す。以下、この図を基に改善案の説明を行う。なお、以降はレコードの位置情報の管理単位のサイズはデータ・ディスクへの分割単位のサイズと同じ１２ＫＢの場合について取り扱う。従来の方式と同じＦＢＡディスクのブロック(セクタ)・サイズと同じ１～２ＫＢでは効率が悪いからである（詳細は、前述した「レコードの位置情報を持つ単位の問題」を参照のこと）。ま

ード・ポインタを使用しない。これにより、レコード・ポインタ領域のサイズは最大でも、１ＣＫＤトラック当り、

４ｂｙｔｅ×９４＝３７６ｂｙｔｅ

で済む。

（２）レコード・ポインタを含むコントロール・フレームのサイズは、別途、制御情報（図２５のコントロール・フレーム・ポインタ）として保持している。この情報により、コントロール・フレームのサイズが可変となっても、制御を複雑にせずに済ませる。

（３）これらの制御情報（コントロール・フレーム）は、各フレームの先頭ではなく、最後尾に記録する。また、コントロール・フレーム内のレコード・ポインタも後ろから順番に並べていく。（実際のレコードがレコード・フレーム内で前にあるものほど、それに対応するレコード・ポインタはコントロール・フレーム内で後ろにくる）。これにより、フォーマット・ライト時の制御も簡単に行える。

【００７６】以下、実際のリード／ライトの動きに従って、フレーム内のレコードの位置決め制御の方法をもう少し詳細に説明する。

（１）リード／アップデート・ライト時

（ａ）物理ディスクからフレームをキャッシュ・メモリ上に読み込んできたら、その最後尾のコントロール・フレームをＣＰＳが読み取る。このとき、コントロール・フレームのサイズは読んでみるまで分からないわけであるが、とりあえず、フレームの最後尾の適当なバイト数をまとめて読んでみる。（このとき読み取るバイト数は、ＣＰＳやキャッシュ・メモリ、バスなどのアーキテクチャにより最適値を選択する必要がある。）例えば、フレームの最後尾３２Ｂを読み取ることにすると、このフレーム内に記録されているレコード数が７個以下の場合は、ＣＰＳはこの１回のアクセスですべてのレコードのレコード・ポインタを取得することができる。例えば４ＫＢサイズをもっとも多いレコードサイズとする場合、１フレーム中のレコード数は３個以下であるから、大部分の場合は、この１回のキャッシュメモリ・アクセスですべてのレコード・ポインタを取得できることになる。

（ｂ）読み取ったコントロール・フレームの最後尾のコントロール・フレーム・ポインタ中のメモリ・アドレスから、コントロール・フレームをまだ読み残しているか　否か、判断できる。もし、コントロール・フレーム

（ｄ）ホストとの間でデータ転送を開始する前に、目的レコードのメモリ・アドレスにカウント・フィールドのサイズ、ＫＬ、ＤＬを加えた値（目的レコードの最後尾のメモリ・アドレス）と、コントロール・フレーム・ポインタ中のコントロール・フレームの先頭を示すメモリ・アドレスを比較する。もし、目的レコードの最後尾のメモリ・アドレスがコントロール・フレームに食い込んでいるような場合は、ＣＰＳはホストとのデータ転送をコントロール・フレームの手前で一端停止し、メモリ・アドレスを次のフレームの先頭まで進めてから、ホストとのデータ転送を再開するような制御を行う。目的レコードの最後尾がレコード・フレーム内に収まっていれば、上記のような制御は不要で、ＣＰＳは、目的レコードのデータ転送を途中で中断することなく完了することができる。

（ｅ）引き続き、ホストからリード／アップデート・ライト系のコマンドが出された場合は、次のレコードはそのままメモリ上で連続しているので、レコード・ポインタを使用しなくてもそのままデータ転送を続けることができる。ただし、レコードの最後尾のメモリ・アドレスと、コントロール・フレームの先頭アドレスの比較は、各レコードごとに行わなくてはならない。

【００７７】（２）フォーマット・ライト時

（２．１）同一フレーム内でレコードを書き足す場合

（ａ）物理ディスクからフレームをキャッシュ・メモリ上に読み込んできたら、その最後尾のコントロール・フレームをＣＰＳが読み取る。

（ｂ）コントロール・フレーム・ポインタを調べ、すべてのレコード・ポインタを取得する。

（ｃ）次に、ホストから送られてくるサーチのアーギュメントと取得したレコード・ポインタから、目的レコードのフレーム上のメモリ・アドレスを見つける（これでオリエンテーションが確立したことになる）。

以上の（ａ）～（ｃ）手順は、リード／アップデート・ライト時と同じである。

（ｄ）ホストから送られてくるＷｒｉｔｅ　ＣＫＤのカウント・フィールドの値からトラック・オーバランを発生しないか否かを調べる。（この詳細な手順は、別途、「トラック容量のチェック方式」として説明する。）トラック・オーバランを発生する場合は、以降の制御でトラック・オーバラン・ポイントまでデータ転送を行い、ホストにトラック・オーバランを報告するような制御を行う。トラック・オーバランのチェックが終ったら、Ｃ

から新しく書き足すレコードの最後尾のメモリ・アドレスを計算し、（ｄ）で更新したコントロール・フレーム・ポインタ中のメモリ・アドレスと比較する。もし、新しいレコードの最後尾のメモリ・アドレスがコントロール・フレームに食い込んでいるような場合は、ＣＰＳはホストとのデータ転送をコントロール・フレームの手前で一端停止し、メモリ・アドレスを次のフレームの先頭まで進めてから、ホストとのデータ転送を再開するような制御を行う。新しいレコードの最後尾がレコード・フレーム内に収まっていれば、上記のような制御は不要で、ＣＰＳは新しいレコードのデータ転送を途中で中断することなく完了することができる。

（ｆ）引き続き、ホストからＷｒｉｔｅＣＫＤコマンドが出された場合は、次のレコードはそのままメモリ上で連続しているので、レコード・ポインタを使用しなくてもそのままデータ転送を続けることができる。ただし、トラック・オーバランのチェック、コントロール・フレーム・ポインタの更新、レコード・ポインタの追加、レコードの最後尾のメモリ・アドレスと、コントロール・フレームの先頭アドレスの比較は、各レコードごとに行わなくてはならない。

（ｇ）コマンドが終了した場合は、ＣＰＳが保持していた更新後のコントロール・フレームをキャッシュ・メモリ上に書き出す。

【００７８】なお、図２６に示すようなケースではレコード・フレームが３バイト空いているからといって、引き続き同じフレーム中にレコードを追加しようとしてレコード・ポインタを追加すると、レコード・ポインタが前のレコードの後部に食い込んでしまうことになる。すなわち、フレームの空き容量が４バイト以下の場合は、同じフレーム中に新たなレコードは追加できず、次のフレームを使用しなくてはならない。このようにして生じてしまった空きバイト（４バイト以下）は、コントロール・フレーム中に含めて管理してしまった方が、（ｅ）のデータ転送時のコントロール・フレームの飛び越しなどがやりやすい。従って、コントロール・フレーム・ポインタ中のメモリ・アドレスはこの空きバイトの先頭アドレスを示すことになる。この空きバイトの先頭から、レコード・ポインタの先頭アドレスまでのオフセット（０～４バイト）は、コントロール・フレーム中の制御情報として保持しておいて、（ｂ）のレコード・ポインタの取得の際、コントロール・フレーム・ポインタ中のメモリ・アドレスからこのオフセット値を引いてレコー

ことになる。従って、この場合は、（２．１）の（ａ）～（ｃ）のコントロール・フレームの取得によるオリエンテーションの確立作業は不要である。（ｄ）のトラック・オーバランのチェックは（２．１）と同様に行う。

（ｅ）トラック・オーバランのチェックが終ったら、ＣＰＳが自分のメモリ上にコントロール・フレーム・ポインタ（コントロール・フレームの先頭アドレスはレコード・ポインタ１個分のところを示す。）を生成し、さらに、新しく書くレコードのレコード・ポインタをＣＰＳが保持しているコントロール・フレームの先頭に書く。

（ｆ）Ｗｒｉｔｅ　ＣＫＤのカウント・フィールドの値から新しく書くレコードの最後尾のメモリ・アドレスを計算し、（ｅ）で生成したコントロール・フレーム・ポインタ中のメモリ・アドレスと比較する。もし、新しいレコードの最後尾のメモリ・アドレスがコントロール・フレームに食い込んでいるような場合は、ＣＰＳはホストとのデータ転送をコントロール・フレームの手前で一端停止し、メモリ・アドレスを次のフレームの先頭まで進めてから、ホストとのデータ転送を再開するような制御を行う。新しいレコードの最後尾がレコード・フレーム内に収まっていれば、上記のような制御は不要で、ＣＰＳは新しいレコードのデータ転送を途中で中断することなく完了することができる。

（ｇ）引き続き、ホストからＷｒｉｔｅ　ＣＫＤコマンドが出された場合は、（２．１）と同様となる。

（ｈ）コマンドが終了した場合は、ＣＰＳが保持していた新しいコントロール・フレームをキャッシュ・メモリ上に書き出す。フォーマット・ライト系の処理では、そのフレームに記録するレコード数が分かってからでないと、コントロール・フレームのサイズが確定しない。従って、コントロール・フレームをフレームの先頭においてしまうと、最初のレコードをフレームのどこから書き始めたらよいのか分からないことになってしまう。

本方式では、上述のようにレコードはフレームの先頭から書き始め、コントロール・フレームはフレームの最後尾から書き始めることにより、このような問題を回避している。

【００８０】次に、この改善案による位置情報のメモリ容量の見積りを行う。この改善案による位置情報のメモリ容量について、前述した「レコードの位置情報のメモリ容量見積り」で見積もったケースのうち図２２の管理単位２のケースについて見積りを行い、改善効果を見る。このケースは、管理単位（フレーム）サイズは１２

は．一つ目のフレームのコントロール・フレームにレコード・ポインタが９４個．コントロール・フレーム・ポインタが１個あって，他のフレームに制御情報は特にない（というより．他のフレームそのものが要らない）。従って、位置情報の制御に必要なメモリ容量は．

４Ｂ×９５＝３８０Ｂ

である。この結果と、「レコードの位置情報のメモリ容量見積り」で見積もった他のケースとの比較を図２７に示す。＜ケース　ｉｉ－２＞は従来の方式で、各ＦＢＡブロック（ここでは、フレーム）の先頭に位置するレコードの位置だけを保持する方式である。＜ケース　ｉｉ－３＞はすべてのレコードの位置情報を保持する方式　＊

＜ケース　Ｉ－１＞　１０２４×４８－４８０４４＝１１０８

＜ケース　Ｉ－２＞　１０２４×４８－４８７９６＝　３５６

＜ケース　Ｉ－３＞　１０２４×４８－４９０２０＝　１３２

この値と図２７の＜ケース　ｉｉ－３＞改の位置情報のメモリ容量３８０Ｂと比較すると、ギャップを全部削除してしまう＜ケース　ｉ－１＞以外ではメモリ容量が足りなくなるように見える。しかし、図２７の値は．レコード数が９４レコードで位置情報（コントロール・フレーム）のメモリ容量がもっとも多くなるときの値であるのに対し、上の値は１トラック１レコードで、ＣＫＤレコードのためのメモリ容量が最大の場合の値である。本改善案では、レコード数が少なくなると、比例してコントロール・フレームのメモリ容量も少なくなり．１トラック１レコードの場合では．２０Ｂで済んでしまう（レコード・ポインタ１個、コントロール・フレーム・ポインタ４個）。結局、本改善方式ではすべての場合について．メモリ容量は足りることになる。（ただし．トラック容量のチェック方式として、＜ケース　ｉｉＩ－３＞．すなわち．各レコードのＣＫＤフォーマットでの相対位置を保持するために．ＥＣＣやパディング等削除した分をギャップを伸ばして調整する方式をとると．９４レコードの場合などメモリ容量が足りなくなる可能性もある。）また．目的レコードの位置決めのためのメモリ・アクセスもフレーム当り１回～２回で済む。このときに一度ＣＰＳがコントロール・フレームをキャッシュ・メモリから取得してしまえば、後の制御ではＣＰＳはこの取得したコントロール・フレームを使用すればよいので制御のためのメモリ・アクセスは不要である。フォーマット・ライトなどでフレーム内のレコードの構成が変わった場合も．ＣＰＳが取得しているコントロール・フレームを更新しておいて　最後にキャッシュ・メモリに

＊で．本による改善前の方式である。＜ケース　ＩＩ－３＞改が本方式である。

【００８１】「レコードの位置情報のメモリ容量見積り」において示したように．ＣＫＤレコードを記録するために正味必要なメモリ容量の最大値は、１トラック１レコード（ＲＯのみ）でそのデータ・フィールドが最長の場合である。このとき必要なメモリ容量は、図２３から引用すると図２８のようになる。ＣＫＤトラック１本のために用意するＦＢＡディスクの容量を４８ＫＢ（１２ＫＢ×４台）とすると．制御情報のために使用できるメモリ容量は下式のようになる。

成したといえる。

【００８２】次に、「セグメント・ナンバを用いたトラック容量のチェック方式の検討」を行なう。ＣＫＤ－ＦＢＡ変換方式を特徴付ける三つのポイントとして．

＜Ｉ＞ギャップの削除方法

＜ＩＩ＞レコードの位置決め方法

＜ＩＩＩ＞トラック容量のチェック方法

をあげたが、ここでは、＜ＩＩＩ＞のトラック容量のチェック方法について、特に＜ケース　ＩＩＩ－３＞のセグメント・ナンバを使用してトラック容量をチェックする方式について．実際の制御方法を説明する。なお、＜ＩＩＩ＞のトラック容量のチェック方法については＜ケース　ＩＩＩ－１＞と＜ケース　ＩＩＩ－２＞と＜ケース　ＩＩＩ－３＞の３つの方式が上がっていた。まず、前提条件を以下のように考える。セグメント・ナンバは．図３５のように、３２Ｂを１セグメントとする。セグメント・ナンバは、１から順番にふられた絶対セグメント番号（ＡＳＮ）を使用するものとする（図２９参照）。また、使用する数字はすべて十進数を用いる。

【００８３】次に、制御方法について説明する。まず基本方針として以下の２点を考える。

（１）ホーム・アドレスＨＡのセグメント・ナンバはＳ／Ｗから読み書きできるので、ＨＡが適宜の位置にある場合（ＡＳＮ＝１７）と．ＨＡがＭｏｖｅしている場合（ＡＳＮ＝２３）の２種類をサポートするものとする。

（２）それ以外のＲＯを含む他のレコードについてはＳ／Ｗからセグメント・ナンバもＳＣ（スキップ・コントロール）も読み書きできないので．ＭｏｖｅやＳｐｌｉ

ライズ時はＡＳＮ＝１７とする。）。
（ｂ）Ｒ０のセグメント・ナンバはＨＡのセグメント・ナンバにかかわらず常にＡＳＮ＝２６とする。
（ｃ）Ｒ１以降のレコード（Ｒｎ）のセグメント・ナンバは下式により定める。
・前のレコード（Ｒｎ－１）にキー・フィールドがある場合．
ＲｎのＡＳＮ＝Ｒｎ－１のＡＳＮ＋［（ＫＬ＋１２）／３２］＋［（ＤＬ＋１２）／３２］＋２２
・前のレコード（Ｒｎ－１）にキー・フィールドがない場合．
ＲｎのＡＳＮ＝Ｒｎ－１のＡＳＮ＋［（ＤＬ＋１２）／３２］＋１５
ただし、１２はＥＣＣとスペースのバイト数、３２はセグメントのバイト数、２２はＧ２（２２４Ｂ）とＧ２（２２４Ｂ）とＧ３（２１６Ｂ）とカウント・フィールド（４０Ｂ）のセグメント数、１５はＧ２（２２４Ｂ）とＧ３（２１６Ｂ）とカウント・フィールド（４０Ｂ）のセグメント数であり、［］内は、小数点以下切上げとする。
【００８４】（２）トラック容量（トラック・オーバラン）のチェック方法
（１）により求めたセグメント・ナンバと、ホストから送られてくるＫＬ，ＤＬを用いて下式によりトラック・オーバランのチェックを行う。下式を満たしていればＯＫ．満たしていなければトラック・オーバランになる。
・追加するレコード（Ｒｎ）にキー・フィールドがある場合．
ＲｎのＡＳＮ＋［（ＫＬ＋１２）／３２］＋［（ＤＬ＋１２）／３２］＋１４≦１５３３
・追加するレコード（Ｒｎ）にキー・フィールドがない場合．
ＲｎのＡＳＮ＋［（ＤＬ＋１２）／３２］＋７≦１５３３
ただし、１４はＧ２（２２４Ｂ）のセグメント数．７はＧ２（２２４Ｂ）のセグメント数、１５３３は最大セグメント数であり．［］内は．小数点以下切上げとする。
【００８５】以上をまとめると、フォーマット・ライト時のトラック・オーバランのチェックの手順は下記のようになる。（１）の手順に従い、書き足すレコードのセグメント・ナンバを求める。
・求めたセグメント・ナンバの値とホストから送られてきたＫＬ．ＤＬから．（２）の手順に従い．書き足すレ

とになる。このような制御にすることにより、トラック・オーバランのチェックでは常に同じ式を用いることができるうえ、また、ＳＣ（スキップ・コントロール）の値を気にする必要が無くなるので、制御が非常にシンプルになる。もし．ＨＡのＭｏｖｅにあわせてＲ０もずらしていると、トラック・オーバランのチェック時．常に．トラックの先頭のＨＡのセグメント・ナンバを読みとってチェックするか、さもなければ．実際のディスクと同じように．各レコードにおいてもＳＣ（スキップ・コントロール）を保持して管理する必要がでてくる。基本的には、フォーマット・ライト時のギャップ中の作業は．（２）のトラック・オーバランのチェックなど、従来のディスク制御に比べ増える方向なので、シンプルにできる部分はシンプルにすべきである。
【００８６】以上より、セグメント・ナンバを使用したトラック容量のチェック＜ケースＩＩＩ－３＞の方法）は比較的シンプルな制御で実現可能である。従って、従来方式＜ケース　ｉｉｉ－２＞の方法のようにＣＫＤフォーマットのトラック・イメージをＦＢＡに変換した後も．ギャップ長を調整して各レコードのトラックの先頭からのバイト位置を、無理に元のＣＫＤトラックのまま維持しなくても制御可能である。セグメント・ナンバ自体はＳ／Ｗインター・フェイス上サポートが必要であるので（少なくともＨＡについては）．それを積極的に利用する＜ケース　ｉｉｉ－３＞の方法は、ギャップ長を元のＣＫＤフォーマットと同等の量だけ保持しなければならない＜ケース　ＩＩＩ－２＞に比べ、特にレコード・サイズが小さいときのキャッシュ・メモリの利用効率において著しく有利である（「ギャップの削除方法とメモリ容量の評価」）。また．すべてのレコードのＫＬ．ＤＬを読んで．トラックの残容量を計算する＜ケースＩＩＩ－１＞に対しては．メモリのアクセス回数の点で有利であり、もっとも有望な方式である。
【００８７】これまでＣＫＤ－ＦＢＡ変換方式を特徴付ける三つのポイントとして．
＜Ｉ＞ギャップの削除方法
＜ＩＩ＞レコードの位置決め方法
＜ＩＩＩ＞トラック容量のチェック方法
をあげ、それぞれについて下記のような検討を行ってきた。
●＜Ｉ＞のギャップの削除方法について
・「ギャップの削除方法とメモリ容量の評価」において．メモリ容量の評価を行った。この結果　すべてのギャッ

ロック内の最初のレコードの位置情報だけ保持する）より有利であるとしていた＜ケース　ＩＩ－３＞（全レコードの位置情報を保持する）は、そのままでは、レコードの位置情報を保持するためのメモリ容量が多くなりすぎることが分かった。

・そこで、「レコードの位置情報の制御方式の改善」で＜ケース　ＩＩ－３＞のメモリ容量の改善が可能か否か検討した。その結果、制御方法を工夫することにより、＜ケース　ＩＩ－３＞でも全レコードの位置情報を保持するためのメモリ容量を十分小さくできることがわかった。

●＜ＩＩＩ＞のトラック容量のチェック方法について

・＜ケース　ＩＩＩ－３＞（セグメント・ナンバなどを用いて各レコードのＣＫＤフォーマットでの相対位置情報を保持する。ギャップが不要な分有利）の実現可能性を「セグメント・ナンバを用いたトラック容量のチェック方式の検討」で検討した。その結果、セグメント・ナンバを用いてトラック容量のチェックを行うことは容易で、その具体的な手順も示した。

【００８８】以上のような検討結果前述した図１９に示すように、もっとも高得点をあげている方式＜１，３，３＞が採用できることがわかった。この方式は、ギャップの削除方法としては全ギャップを削除するので、特にジャーナル・ファイルなどのレコード・サイズが小さい場合、トラック・イメージを格納するためのメモリ容量が少なくて済む。レコードの位置決め方式としては、全レコードの位置情報を保持する方式を採用しているので、目的レコードを探すためのメモリ・アクセス回数が少なくて済む。ただし、位置情報を保持するためのメモリ容量を少なくするために、「レコードの位置情報の制御方式の改善」で検討した＜ケース　Ｉｉ－３＞改を採用する必要がある。トラック容量のチェック方式としては、セグメント・ナンバを使用する方式なので、各レコードの相対位置をＣＫＤフォーマットのまま維持するためのギャップは必要無く、従って、ギャップの削除方法として、全ギャップを削除する方法との併用が可能となる。単に得点の大小だけで判定するのでなく、高得点をあげている他の方式についても、方式＜１，３，３＞との比較を行っておく。

【００８９】＜２，３，２＞　・・・５４点

図９において次点である方式＜２，３，２＞は、＜１，３，３＞と比較するとトラック容量のチェック方法でセグメント・ナンバを使用せず、従来方式であるギャップ

トラック容量のチェックをしなくてもよいことである。従って、セグメント・ナンバそのものも省略できる。悪い点は、ギャップ長を伸ばして各レコードのＣＫＤフォーマットでの相対位置を維持するため、メモリ上でのトラック容量は常にほぼＣＫＤトラック１本分必要となることである。これは、特に、ジャーナル・ファイルなどのレコード・サイズの小さい場合にキャッシュ・メモリの利用効率悪化、ステージング／デステージング時の内部バスの負荷増大につながる。また、レコードの位置決め方式として、全レコードの位置情報を保持する方式を併用しているが、レコードサイズが最少で、レコード数が最多の９４レコード／トラックのケースでは、「セグメント・ナンバを用いたトラック容量のチェック方式の検討」で検討した改善方法をとっても、メモリ容量不足となる可能性もある。また、この方式は、レコードの位置決め方式は従来方式と異なるが、トラック容量のチェック方式は従来方式そのままである。

【００９０】＜２，３，３＞　・・・５３点

この方式は、ギャップを一部残すこと以外は＜１，３，３＞と同じである。従って、ギャップを残しても特にそれを使用しないので、ギャップを残す意味がなく、＜１，３，３＞に比べ、わざわざ採用する必然性がない方式である。上記説明では、アレイ型ディスク装置（ＲＡＩＤ）を前提としていたが、ここでは、通常のディスク装置を用いる場合にも＜１，３，３＞の方式を採用できることについて述べる。

●＜Ｉ＞のギャップの削除方法について

ギャップの削除方法によるメモリ容量の大小がキャッシュの利用効率に大きく関わることは、ＲＡＩＤであってもなくても同じであり、この点からは、すべてのギャップを削除する＜ケース　Ｉ－１＞の方式が有利なことに変わりはない。

●＜ＩＩ＞のレコードの位置決め方法について

バスの負荷を下げ高転送レートを出しやすくするためには、１回のバス獲得である程度まとまった量のデータ転送を行ってしまう方がよい。すなわち、レコードの位置情報も一括してアクセスできる方が、毎回各レコードのカウント・フィールドをアクセスする方式よりも有利である。従って、レコードの位置決め方法についても、メモリ・アクセス回数が少ない＜ケース　ＩＩ－３＞（全レコードの位置情報を保持する）の方が、従来方式の＜ケース　ＩＩ－２＞（ＦＢＡブロック内の最初のレコードの位置情報だけ保持する）より有利である。

て各レコードのＣＫＤフォーマットでの相対位置情報を保持する。ギャップが不妥な分有利）のままでよい。

【００９１】次に、ＣＫＤ－ＦＢＡ変換におけるコマンド・エミュレーション方式について検討する。ここでは、前述した具体的な変換方式に基づいて、実際のＳ／Ｗインタフェース上の代表的な、コマンドごとにこのＣＫＤ－ＦＢＡ変換方式における実現方法を検討する。

【００９２】１．ＳＥＴ　ＳＥＣＴＯＲ

＜機能＞チャネルから１バイトのセクタ情報を受け取り、セクタを検索する。

（セクタ情報：Ｘ' ００' ～Ｘ' ＤＤ、Ｘ' ＦＦ'）

＜実現方式＞このディスク・サブシステムでは、セクタ値は、まず、ＣＫＤトラックを四つに分割したフレームのうち、どのフレームに目的レコードが存在するかを知るために使用する。フレームが見つかった後は、コントロール・フレーム中のレコード・ポインタに書かれている各レコードのセグメント・ナンバと、このセクタ値を比較することにより、より精度の高い位置決めができる（フレーム、及びコントロール・フレームやレコード・ポインタの概略については図２５に示したが、詳細は「レコードの位置情報の制御方式の改善」参照のこと）。

○目的レコードが存在するフレームの計算

実際のＣＫＤディスクでは、セクタ１個分は２２４バイトからなりたっているので、セクタ値からＣＫＤフォーマットのトラック上でのそれに相当する絶対バイト・アドレスはただちに計算できる。しかし、フレーム上ではギャップやＥＣＣ、パディングなどは削除して記録しているので、その絶対バイト・アドレスがどのフレームに含まれているかは、その削除してしまった分を換算しないと分からない。この削除した分量はレコード数にも依存する。

【００９３】以下、目的レコードが存在するフレームの計算について説明する。フレーム上ではＣＫＤフォーマットのトラックから何バイト削除しているかは、フレーム上でのトラック・フォーマットが詳細に決らなくては定まらないが、図３３のようなフォーマットであるとすると、各フィールドごとにフレーム上で削除されているバイト数は図３１のようになる。目的レコードの前にはＲ０を除いて（目的レコードのレコード・ナンバ－１）個のレコードが存在するので、セクタ値で示されるＣＫＤフォーマットのトラック上での位置は、フレーム上ではトラックの先頭から下式のバイト数ＢＣ１で表される

バイト数ＢＣ２は下式のようになる。

ＢＣ２＝ＢＣ１＋４×レコード・ナンバ

フレーム１個にはコントロール・フレーム・ポインタ４バイトを除くと、レコード・ポインタも含め１２ＫＢ－４Ｂ記録できるので、セット・セクタ値に相当する位置は、下式で求められるＮ番目のフレームに存在することが期待できる。

Ｎ＝［ＢＣ２／１２２８４］

ただし、［］内は、小数点以下切上げとする。これらの式を一つにまとめると、下式のようになる。

Ｎ＝［（２２４×セクタ値－２８６－７４８×レコードナンバ）／１２２８４］

ただし、［］内は、小数点以下切上げとする。また、負の値になったときはＮ＝１とする。正確には、フレーム上で削除されているバイト数は各レコードにキー・フィールドが存在しているか否か、また、３２の整数倍にするためにＫＬ、ＤＬにゼロ・パディングしているバイト数などにもよって異なってくるため、上式は、あくまで概算式となる。上式では、このような不定の部分については、削除されるバイト数が最大になるように仮定して計算しているので、これにより求めたフレームの番号は実際よりも前の方を示す傾向になる。従って、上式で求めたフレームに目的レコードが存在しなかった場合は、次のフレームを探せばよい。しかし、この式の精度はもともとフレームのサイズ１２ＫＢでよく、それに対し誤差の量はレコード１個当り最大でも２８６バイト（２２４＋３１＋３１）であるので、通常は問題無い。

【００９４】また、ＳＥＡＲＣＨコマンドが来なかったり、ＳＥＡＲＣＨコマンドであってもＳＥＡＲＣＨ　ＫＥＹであったためレコードナンバが得られなかった場合は、フレームを特定することを諦めて、任意のフレーム（例えば最初にステージングが完了したフレーム）中のレコードのセグメント・ナンバを調べていくことになる（次の項の手順参照）。

○目的レコードの探索

上の手順により、目的レコードが存在するフレームを特定したら、後は、そのフレーム中のレコード・ポインタを調べて、目的レコードを探せばよい（詳細な手順は、「レコードの位置情報の制御方式の改善」参照）。まず、そのフレームの先頭のレコードのレコード・ポインタ（コントロール・フレーム中では最後尾にある）から順番にレコード・ポインタ中に記録されているセクタ値を調べていく。そして、チャネルから送られてきたセク

ードのカウント・フィールドを比較し．一致していればそのレコードにオリエンテーションを確立する。もし、一致しなければ次のレコードのカウント・フィールドとの比較を繰り返していき．最初にアーギュメントと一致したレコードにオリエンテーションを確立し、次のコマンドに移っていく（ＣＫＤの論理トラックの最後まで探して一致するレコードが見つからなかった場合などについては、「４．ＳＥＡＲＣＨ　ＩＤＥＱ」コマンドを参照のこと）。なお、先行するＳＥＥＫコマンドによるヒット・チェックの結果、既に目的トラックがキャッシュ上に存在することが分かっている場合（ヒット時）はただちに次のコマンドへ進んでよいが．ミス・ヒット時、または、ヒットしていてもトラックの一部分しかキャッシュ上に存在しておらず．目的レコードが含まれるフレームがキャッシュ上に存在しない場合は、チャネル・エンドＣＥのみ返し一旦チャネルからディスコネクトして．目的のフレームを物理ディスクからステージングした後．デバイス・エンドＤＥを返して次のコマンドへ進むことになる。

【００９５】２．ＳＥＡＲＣＨ　ＩＤＥＮＴＩＦＩＲＥＱＵＡＬ

＜機能＞チャネルから５バイトのＳＥＡＲＣＨ情報を受け取り、デバイスから読み出されたカウント・フィールド（Ｒ０フィールドも含む）のＣＣＨＨＲと比較する。

＜実現方式＞

○ＳＥＴ　ＳＥＣＴＯＲコマンドからチェインした場合

「１．ＳＥＴ　ＳＥＣＴＯＲ」で示した動作により、ＣＣＨＨＲの一致したレコードにオリエンテーションを確立する。実際のＣＫＤディスクでは．ＳＥＴ　ＳＥＣＴＯＲ後、最初に見つけたレコードのＣＣＨＨＲが一致しなければ不一致報告をし（ＣＥ，ＤＥ）、ホストがトランスファーインチャネルＴＩＣを併用して再度ＳＥＡＲＣＨ　ＩＤ　ＥＱコマンドを発行するのを待つ。しかし．このサブシステムでは．「１．ＳＥＴ　ＳＥＣＴＯＲ」で示したように一回のＳＥＡＲＣＨ　ＩＤ　ＥＱコマンドでトラック全体をサーチできてしまう。エミュレーションの厳密性からいえば、あくまで、１個のＳＥＡＲＣＨ　ＩＤ　ＥＱコマンドで１個のレコードのサーチしか行うべきでないが、そうすることによるメリットは特に考えられないので、処理効率の点から「１．ＳＥＴ　ＳＥＣＴＯＲ」で示した一括サーチの方式でよいと考える。ＳＥＴ　ＳＥＣＴＯＲで指定されたセクタから後ろに一致するレコードが無かった場合は．マルチ・トラ

初のフレームの最初のレコード（Ｒ０も含む）に戻って比較を繰り返し．それでも一致するレコードが見つからなければＮｏ　Ｒｅｃｏｒｄ　Ｆｏｕｎｄを報告する。

○ＲＥＡＤ，ＷＲＩＴＥ．ＳＥＡＲＣＨ系コマンド、または．ＳＰＡＣＥ　ＣＯＵＮＴコマンドからチェインした場合

先行するコマンドでオリエンテーションしていた次のレコードに対しＣＣＨＨＲの比較を行う。この場合は、既にオリエンテーションが確立しているのであるから、ＳＥＴ　ＳＥＣＴＯＲコマンドからチェインした場合と異なり．実物のＣＫＤディスク同様、サーチするレコードは次のレコード１個だけの方がよい。オリエンテーションしていた所より後ろにレコードが存在していなかった場合は、マルチ・トラック・ビットがセットされていれば次のトラックに移り、セットされていなければ同じトラックの先頭に戻って、比較する。ただし、同一ＣＣＷチェイン内でもう一度、トラックの最後までいってしまったらＮｏ　ＲｅｃｏｒｄＦｏｕｎｄを報告する。

○他のコマンドからのチェイン、または、チェインの先頭であった場合

この場合、実際のＣＫＤディスクでは．現在のトラック上で最初に発見したレコードに対しサーチ動作を行うが．このサブシステムでは常にＲ０から順番にサーチするほかは、「ＳＥＴ　ＳＥＣＴＯＲコマンドからチェインした場合」と同じでよい。ミスヒットしていてステージングが完了していなかった場合や．マルチ・トラック・ビットがたっていて、次のトラックがミスヒットしていた場合などは．このコマンドでステージングを完了させる。また、ＳＥＥＫコマンドが先行しなかった場合の現在トラックの認識のためにそのボリュームに対する最後のアクセス・トラックは常に覚えておかなければならない。

【００９６】３．ＲＥＡＤ　ＤＡＴＡ

＜機能＞ディスクから読み出したデータ・フィールドの情報をチャネルに転送する。

＜実現方式＞

○ＲＥＡＤ，ＷＲＩＴＥ．ＳＥＡＲＣＨ系コマンド、または．ＳＰＡＣＥ　ＣＯＵＮＴコマンドからチェインした場合

フレーム上でオリエンテーションが確立していたフィールド以降にある最初のデータ・フィールドの情報をチャネルに転送し．オリエンテーションを維持する。もし、オリエンテーションが確立していたフィールドより後ろ

比較を満足したＳｅａｒｃｈ系コマンド、または、ＲＯＣをバイパスしたＳＰＡＣＥ　ＣＯＵＮＴコマンドからチェインした場合のみである）。

○他のコマンドからのチェイン、または、チェインの先頭であった場合

オリエンテーションが確立していない状態であるので、実際のＣＫＤディスクでもどのレコードを検出するかは一定でない。従って、毎回、Ｒ１（Ｒ０の次のコマンド）のデータ・フィールドの情報を転送して特に問題ない（Ｒ１が存在すれば）。転送後は、そのままオリエンテーションを確立する。もし、Ｒ１が存在せず、かつ、マルチ・トラック・ビットがセットされていれば、次のトラックのＲ１のデータ・フィールドを転送する（アドレス・マークが見つからないまま、インデックスを検知してしまった場合に相当する）。マルチ・トラック・ビットがセットされていなければ、Ｎｏ　Ｒｅｃｏｒｄ　Ｆｏｕｎｄを報告する。もちろん、次のトラックがキャッシュ上に存在しなかった場合は、まず、ステージングしてからチャネルへデータ転送する。同一ＣＣＷチェイン内でＳＥＥＫコマンドが先行していない場合は、同一ボリュームにおける最後にアクセスしたトラックに対して、ＲＤ　ＤＡＴＡコマンドを実行する。

【００９７】４．ＷＲＩＴＥ　ＣＯＵＮＴ　ＫＥＹ　ＡＮＤ　ＤＡＴＡ

＜機能＞チャネルから受け取った情報を、Ｒｎフィールド（Ｒ０フィールドを除く）としてディスク上に書き込む。

＜実現方式＞このコマンドは、ＣＣＨＨＲの一致したＳＥＡＲＣＨ　ＩＤ　ＥＱコマンド（ＲＤ　Ｄ，ＲＤ　ＫＤ，ＷＲ　Ｄ，またはＷＲ　ＫＤが続いていてもよい），すべてのキーの一致したＳＥＡＲＣＨ　ＫＥＹ　ＥＱコマンド（ＲＤ　Ｄ，またはＷＲＤが続いていてもよい），ＷＲ　Ｒ０コマンド，またはＷＲ　ＣＫＤコマンドからチェインした場合以外はコマンド・リジェクトとなるので、このコマンドが実行される場合は事実上、必ず既にオリエンテーションが確立している。従って、チャネルから情報を受け取ったら、ただちに下記のようなオペレーションを実行する（図２５参照）。

（１）カウント・フィールドの受取

チャネルからＲｎのＣＣＨＨＲＫＬＤＬを受け取り、それに、R_{n-1}のカウント・フィールドの他の情報をつけたす。まだ、キャッシュ上には書き出さない。Ｒｎのカウント・フィールドの他の情報としては、ＳＣ（スキ（計算方法は「セグメント・ナンバを用いたトラック容量のチェック方式の検討」参照）。また、同時にこの値からセクタ値も計算して、コントロール・フレーム中のレコード・ポインタに（３）で書き込む。ＰＡ，ＦはＨＡと同じ値のままでよい。

（２）トラック・オーバランのチェック

チャネルから受け取ったＫＬ，ＤＬとセグメント・ナンバからからトラック・オーバランを発生しないか否かを調べる（詳細な計算方法は「セグメント・ナンバを用いたトラック容量のチェック方式の検討」参照）。もし、トラック・オーバランを発生する場合は、以降の制御でトラック・オーバラン・ポイントまでデータ転送を行い、ホストにトラック・オーバランを報告するような制御を行う。

（３）レコード・ポインタの追加

トラック・オーバランのチェックが終ったら、現在オリエンテーションを確立しているフレームの（ＣＰＳが自分で保持している）コントロール・フレーム・ポインタ中のコントロール・フレームの先頭を示すメモリ・アドレスを４Ｂ減じ（コントロール・フレームを４Ｂ拡張する），さらに，Ｒｎのレコード・ポインタを（ＣＰＳが保持している）コントロール・フレームの先頭に書き足す。もし、４Ｂ減じることによって、コントロール・フレームがレコード・フレームに食い込んでしまうような場合は、そのフレームにレコードを追加することは諦め，次のフレームのコントロール・フレームにレコード・ポインタを追加する。必要ならば，マネージャに次のフレームのエリアを確保してもらう。

（４）フレームのさかいめのチェック

チャネルから受け取ったＫＬ，ＤＬからＲｎの各フィールドの最後尾のメモリ・アドレスを計算し、（３）で更新したコントロール・フレーム・ポインタ中のメモリ・アドレスと比較する。もし，Ｒｎのどれかのフィールドの最後尾のメモリ・アドレスがコントロール・フレームに食い込んでいるような場合は、ホストとのデータ転送をコントロール・フレームの手前で一端停止し，メモリ・アドレスを次のフレームの先頭まで進めてから，ホストとのデータ転送を再開するような制御を（ＣＰＳが）行う。Ｒｎの最後尾がレコード・フレーム内に収まっていれば、上記のような制御は不要で，Ｒｎのデータ転送を途中で中断することなく実行することができる。

（５）カウント・フィールドのキャッシュへの書きだし

これらの一連のチェックが完了してからＣＰＳ内のカウ

（4）のチェックの結果．Rnの最後尾がレコード・フレーム内に収まっていれば．まず、キー・フィールドのデータをチャネルから受けとって、キャッシュ上に書き込んだのち、適当なギャップ相当時間経過後、データ・フィールドのデータもチャネルから受けとってキャッシュに書き込む。もし、（4）のチェックの結果．途中でフレームの切れ目に遭遇することが分かっていれば、そこで一旦データ転送を中断する。現在オリエンテーションが確立しているフレームが最終フレームでなかった場合は．CPSはその次のフレームのキャッシュ上でのアドレスも知っているので．データ転送のためのメモリ・アドレスを次のフレームの先頭に進めた後、チャネルとのデータ転送／キャッシュへの書き込みを再開する。現在オリエンテーションを確立しているフレームが最終フレームであった場合は、もともとキャッシュ上にはその次以降のフレームの領域は確保されていないので．CPSはマネージャに次のフレームのための領域を要求して．そのアドレスをマネージャから教えてもらった後、そこに対してデータ転送を再開する（このような新しいフレームの領域確保などの処置はギャップ相当のタイミングで、データ転送を中断している間に実行してしまえばよい）。データ転送したフレームに対しては．各フレームのコントロール・フレーム中の相当するFBAブロックの更新フラグをセットしておく。

（7）CEの報告

ここまですんだら、CPSはコマンド・チェインの有無を確認するため．一旦チャネルにチャネルとの間のデータ転送が終了したことを示すチャネル・エンドCEを報告する。

－－－コマンド・チェイン指示がなかった場合

（8）CPS→マネージャ　デステージング要求

コマンド・チェイン指示がなければ．CPSはRnの最後尾が書かれたフレームの残りの部分にゼロ・パディングするとともに．ゼロ・パディングをした部分に対応するFBAブロックの更新フラグもセットする。次いで、更新したフレームのコントロール・フレームをキャッシュ上に書き出した後、マネージャを介しDPSにデステージングを要求する。この際、Rnの最後尾が書かれたフレームより後ろにもともとの最終フレームあった場合は．CPSはマネージャにRnの最後尾のフレーム以降のフレームが無効になったことも報告する。もともとRnの最後尾が書かれたフレームより最終フレームが前であった（同じ場合も含む）場合は．それ以降のフレーム

テナンスする）．DPSに該当トラックのデステージングを指示する。この際マネージャは．DPSに対し、物理ディスクに書き出すべきデータのアドレスとして、キャッシュ上での有効データの存在するフレームのアドレスを知らせるとともに、新たに無効フレームが発生した場合のみ、無効フレームに書き込むデータとして．無効フレーム用のデータをあらかじめどこか一箇所に用意しておき、そこのアドレスを指示する（すなわち．無効フレームはすべて無効フレーム用のデータが用意されているメモリ上の同一のデータを物理ディスクに書き込むことになる。この無効フレーム用データはキャッシュ・メモリ上にマネージャが用意しておいてもよいし．各DPSが自分のボードの中に持っていてもよい）。新たに無効フレームが発生しなかった場合は．DPSに対しもともと無効フレームであったフレームをあらためて書き直すことは指示しない。

（10）デステージング

DPSはマネージャからの指示を受け．キャッシュ・メモリ上のデータを物理ディスクに書き出す。DPSはマネージャから指示されたフレームの更新フラグがたっているFBAブロックについてライトを行う。

（11）デステージング終了報告

物理ディスクへの書き出しが終了すると、DPSはマネージャを介して．CPSに終了報告を行う。

（12）DE報告

CPSはチャネルに、全ての処理が終了したことを示すデバイス・エンドDEを報告する。

－－－コマンド・チェイン指示があった場合

（8）' コマンド・チェインが指示されれば、CPSはDEを報告し．次のコマンドを受け取る。

（9）' 次のコマンドがWR　CKDまたはERASEであればその処理に移る。他のコマンドであれば．CPSはCE，UCK，SMを報告しコマンド・リトライを要求する。

（10）' この後、上記の（8）～（11）を実行する。

（11）' DPSから物理ディスクへのライト終了が報告されたら、CPSはDEを報告し．先にコマンド・リトライ要求したコマンドを受け取り直し、そのコマンドの実行に移る。

【0098】5．WRITE　DATA

＜機能＞チャネルから受け取った情報により、レコードのデータ・フィールドを更新する。

レーションを実行する。

（1）フレームのさかいめのチェック

データを書き込もうとしているレコードのカウント・フィールドのKL，DLからRnの最後尾のメモリ・アドレスを計算し，現在オリエンテーションを確立しているフレームのコントロール・フレーム・ポインタ中のメモリ・アドレスと比較する。もし，Rnの最後尾のメモリ・アドレスがコントロール・フレームに食い込んでいるような場合は，ホストとのデータ転送をコントロール・フレームの手前で一端停止し，メモリ・アドレスを次のフレームの先頭まで進めてから，ホストとのデータ転送を再開するような制御を（CPSが）行う。Rnの最後尾がレコード・フレーム内に収まっていれば，上記のような制御は不要で，Rnのデータ転送を途中で中断することなく実行することができる。

（2）データ・フィールドの受取

（1）のチェックの結果，Rnの最後尾がレコード・フレーム内に収まっていれば，データ・フィールドのデータをチャネルから受けとってキャッシュに書き込む。もし，（1）のチェックの結果，途中でフレームの切れ目に遭遇することが分かっていれば，そこで一旦データ転送を中断する。アップデート・ライトの場合は，このようなケースでは必ず次のフレームは存在しているので，CPSはデータ転送のためのメモリ・アドレスを次のフレームの先頭に進めた後，チャネルとのデータ転送／キャッシュへの書き込みを再開する。データ転送したフレームに対しては，各フレームのコントロール・フレーム中の相当するFBAブロックの更新フラグをセットしておく。

（3）CEの報告

ここまですんだら，CPSはコマンド・チェインの有無を確認するため，一旦チャネルにチャネルとの間のデータ転送が終了したことを示すチャネル・エンドCEを報告する。

－－－コマンド・チェイン指示がなかった場合

（4）CPS→マネージャ　デステージング要求

コマンド・チェイン指示がなければ，マネージャを介しDPSにデステージングを要求する。アップデート・ライトの場合は，新たに無効フレームが発生するようなことはない。

（5）マネージャ→DPS　デステージング要求

マネージャは，DPSに対し，物理ディスクに書き出すべきデータのアドレスとして，キャッシュ上での有効デ

（7）デステージング終了報告

物理ディスクへの書き出しが終了すると，DPSはマネージャを介して，CPSに終了報告を行う。

（8）DE報告

CPSはチャネルに，全ての処理が終了したことを示すデバイス・エンドDEを報告する。

－－－コマンド・チェイン指示があった場合

（4）' コマンド・チェインが指示されれば，CPSはDEを報告し，次のコマンドを受け取る。

（5）' 次のコマンドがSEARCH系のコマンド，または，フォーマット・ライト系コマンドであればその処理に移る。同一のコマンドチェイン内で，SEARCH系またはWRITE系以外のコマンドが指示された場合は，CPSはCE，UCK，SMを報告しコマンド・リトライを要求する。

（6）' この後，上記の（4）～（7）を実行する。

（7）' DPSから物理ディスクへのライト終了が報告されたら，CPSはDEを報告し，先にコマンド・リトライ要求したコマンドを受け取り直し，そのコマンドの実行に移る。

【0099】以上のように，この実施例によれば，可変長レコードの1トラックからレコード間ギャップ及びフィールド間ギャップ等全てのギャップを削除し，そのトラックデータを固定長レコードの固定ブロック長サイズの整数倍に相当する管理単位に分割し，その管理単位に含まれる全ての可変長レコードの位置を示す制御情報を上記管理単位の最後尾に後ろから順番に書き入れた後，固定長レコードの固定ブロック長サイズに分割すると共に，各固定長レコードには，可変長レコードフォーマットでのトラックの先頭からの相対位置を示す情報を保持させるので，固定ブロック内で目的の可変長レコードの位置確認のためのメモリアクセス回数が少なくて済みメモリ負荷軽減となると共に，全てのギャップを削除できるのでトラックデータを保持するためのメモリ容量が少なくて済み，従って，ディスクキャッシュをベースとしたディスクサブシステムに好適で，キャッシュのヒット率が向上する。さらに，制御情報領域のサイズを可変にでき可変長レコードの個数だけ用意すればよくメモリが少なくて済む。

【0100】実施例2．上記実施例1においては，図25に示したようにコントロールフレームの後ろにもたせコントロールフレームの長さを可変とする場合を示したが　図24に示すように位置情報領域すなわちコントロ

りコントロールフレームのサイズが固定されると共にレコードフレームのエリアも固定されるため、管理単位のアクセス方法が単純になる。

【０１０１】実施例３．上記実施例１においては、可変長レコードフォーマットでのトラックの先頭からの相対位置として、セクタ値を用いる場合を示したが、これはセットセクタコマンドで用いられる相対位置がセクタ値であることからセクタ値を使用することが望ましいためである。もし、セットセクタコマンドがその他の相対位置例えば、可変長レコードフォーマットでのトラックの先頭からのセグメント値を用いている場合には、この相対位置はセクタ値ではなくセグメント値を用いても構わない。すなわち、図２５に示したレコード・ポインタのセクタ値という値は、セグメント値であっても構わない。

【０１０２】実施例４．上記実施例１においては、トラック容量を計算する場合にはＣＫＤレコード各々にセグメント値を持たせている場合を示したが、トラック容量をセグメント単位に管理するのではなく、セクタ単位に管理する場合にはセグメント値ではなくセクタ値を持たせておいても構わない。

【０１０３】実施例５．上記実施例１においては、コントロール・フレーム内のレコード・ポインタにセクタ値を持たせ、レコード・フレーム内の各ＣＫＤレコード内にはセグメント値を持たせる場合を示したが、セクタ値とセグメント値は一定の関係にあり（セクタ値×？＝セグメント値）、どちらか一方を用いる場合でも構わない。すなわち、レコード・ポインタにセクタ値を有しておれば各ＣＫＤレコードにセグメント値は覚えておかなくてもよい。ただし、この場合はトラックの使用容量がセクタ単位しか計算できなくなる。一方、レコード・ポインタにセクタ値を有さず、レコード・フレーム内のＣＫＤレコードにセグメント値を持たせている場合には、トラックの使用容量がセグメント単位で計算することが可能になる。しかしこの場合には、セットセクタコマンドでセクタ値が指定された場合にレコード・ポインタには、セクタ値が存在しないためレコード・ポインタのメモリアドレスからＣＫＤレコードを探し出し、そのＣＫＤレコードにあるセグメント値からセクタ値を算出しなければ、検索しているセクタかどうかが判定できなくなる。

【０１０４】実施例６．上記実施例１では特に明記しなかったが、ＦＢＡディスク９が１つのディスクで構成されＣＫＤ－ＦＢＡ変換方式が磁気ディスク制御装置５に存在する場合を示したが、ＣＫＤ－ＦＢＡ変換を行うのは磁気ディスク制御装置である場合に限らず、ホスト計算機１のチャネルが行っても構わない。あるいは磁気ディスク装置等の記憶装置自身が行っても構わない。図３２は、磁気ディスク装置等の記憶装置自身がＣＫＤ－ＦＢＡ変換を行う場合の一実施例を示す図である。図３２に示すように、ＣＫＤの磁気ディスク制御装置５００のキャッシュ・メモリに記憶された可変長レコードを変換して、固定長ブロックをもつＦＢＡディスク９００に記憶し直す場合には、ＦＢＡディスク９００内にあるＣＫＤ－ＦＢＡ変換部１００が動作する。ギャップ削除部１０１は、磁気ディスク制御装置５００のキャッシュ・メモリに記憶された可変長レコードをキャッシュ・メモリ６００に読み込む。この場合は、磁気ディスク制御装置５００のキャッシュ・メモリの１トラックのレコードを全てキャッシュ・メモリ６００に読み込む。そしてこの１トラック分の可変長レコードからレコード間ギャップ及びフィールド間ギャップを削除する。キャッシュ・メモリ６００に読み込まれたＣＫＤのトラックイメージ７００は、この時点でギャップが削除されたトラックデータ７ａに変換される。

【０１０６】次に、レコード配置部１０２はトラックデータ７ａをフレームと呼ばれる管理単位７ｂ，７ｃ，７ｄ，７ｅに順に配置する。このフレームと呼ばれる管理単位７ｂ～７ｅのサイズは例えばＣＫＤのトラックイメージ７が４８ＫＢのサイズをもっている場合には４等分した１２ＫＢのサイズをそれぞれ有している。レコード配置部１０２はトラックデータ７ａの可変長レコードを順にフレーム７ｂから配置してゆく。この場合に、各フレームに含まれる可変長レコードの位置を示すアドレス情報を各フレームに記憶する。また、そのレコードがＣＫＤディスク１０で記憶されていたトラック先頭からの相対位置（セクタ値）を示すセクタ情報を同様に記憶する。

【０１０７】次に、固定ブロック記憶部１０３はフレームをＦＢＡディスク９００の記憶部２００に記憶する。ここでＦＢＡディスク９の固定長ブロックのサイズをそれぞれ１ＫＢとすると、フレーム７ｂ，７ｃ，７ｄ，７ｅはそれぞれ１２ＫＢのサイズをもっているため、１つのフレームは１２個の固定長ブロックを用いて記録される。

【０１０８】このようにして、ＣＫＤの磁気ディスク制

きる。また、上記位置情報により目的とするＣＫＤレコードの位置確認ができるのでメモリアクセス回数及びメモリ容量の低減に役立つ。

【図面の簡単な説明】

【図１】この発明のフォーマット変換方式の概略ブロック図である。

【図２】この発明のフォーマット変換方式を説明するためのブロック図である。

【図３】この発明のフォーマット変換の動作を説明する図である。

【図４】この発明のフォーマット変換の動作を説明する図である。

【図５】この発明のフォーマット変換の動作を説明する図である。

【図６】ギャップの残し方を説明を説明するための図である。

【図７】レコードの位置決め方法を説明するための図である。

【図８】トラック容量のチェック方法を示す図である。

【図９】ＣＫＤ－ＦＢＡ変換方式の比較表を示す図である。

【図１０】ＣＫＤレコードのＦＢＡブロックへの分割例を示す図である。

【図１１】ＲＡＩＤにおけるＣＫＤトラックの分割例を示す図である。

【図１２】メモリ容量の見積りを示す図である。

【図１３】トラック当りのレコード数を示す図である。

【図１４】トラック当りのレコード数を示す図である。

【図１５】トラック当りのレコード数を示す図である。

【図１６】トラック当りのレコード数を示す図である。

【図１７】メモリ容量の見積りを示す図である。

【図１８】メモリ容量の見積りのグラフ図である。

【図１９】１トラック分のイメージデータが少ない場合のメリットを示す図である。

【図２０】レコードポインタの保持情報を示す図である。

【図２１】位置情報の管理単位を示す図である。

【図２２】位置情報のメモリ容量の見積りを示す図である。

【図２３】レコードのために必要な最大メモリ容量を示す図である。

【図２４】位置情報の保持方法の一例を示す図である。

【図２５】レコード位置決め方式の改善案を示す図である。

【図２９】Ｅ１８８０系のセグメントナンバを説明する図である。

【図３０】仮想ＣＫＤトラック上のフォーマットイメージを示す図である。

【図３１】フィールド毎の最大削除バイト数を示す図である。

【図３２】この発明の他の実施例を示す図である。

【図３３】ＣＫＤレコードのフィールドの内容を示す図である。

【図３４】サーボ面サーボ方式を用いる磁気ディスクの概略図である。

【図３５】トラックとセクタとセグメントの関係を示す図である。

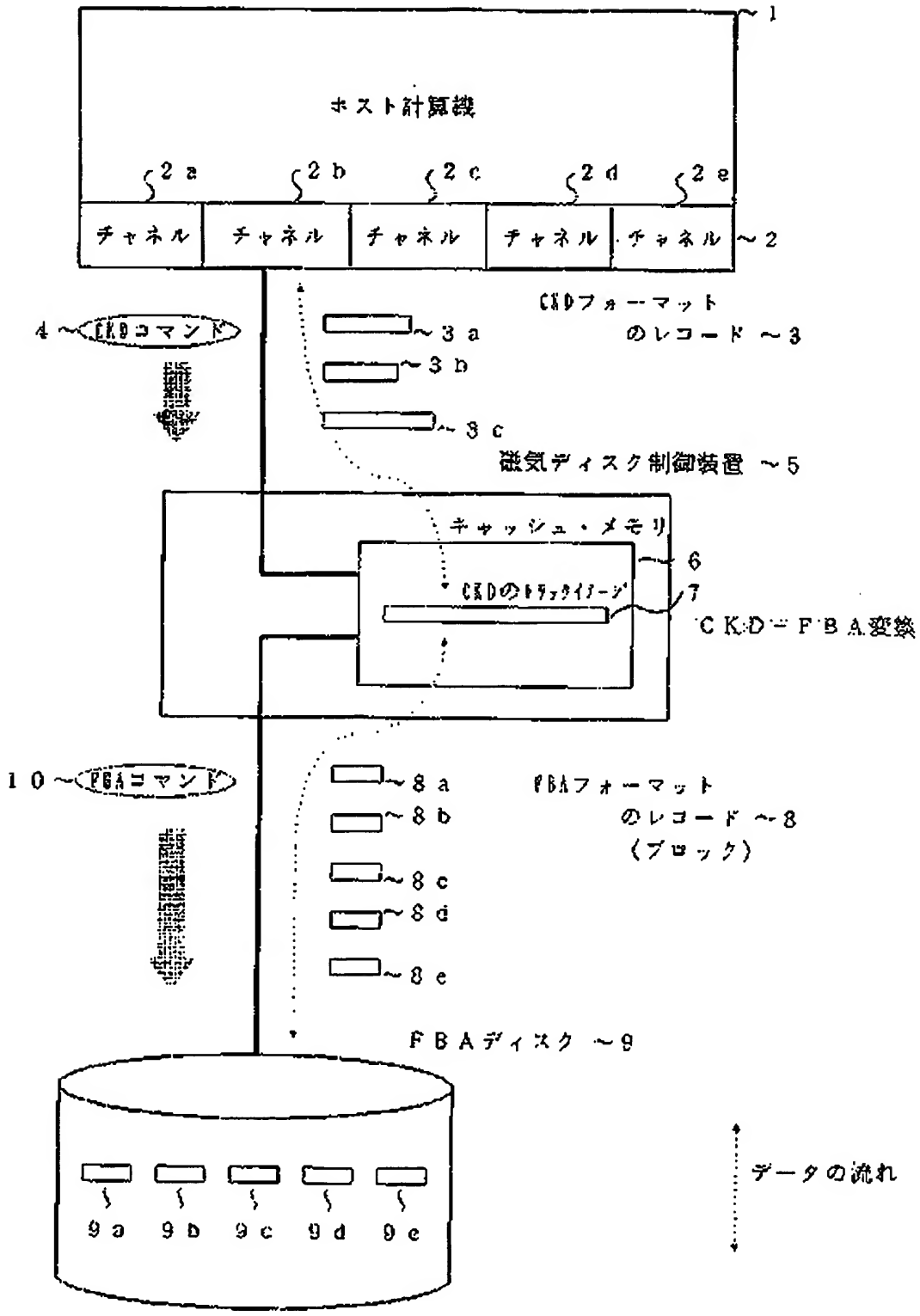
【図３６】従来のソフトウェアがＣＫＤレコード方式に基づいて出力するコマンドシーケンスの一例を示す図である。

【図３７】従来のＣＫＤ－ＦＢＡ変換方式を示す図である。

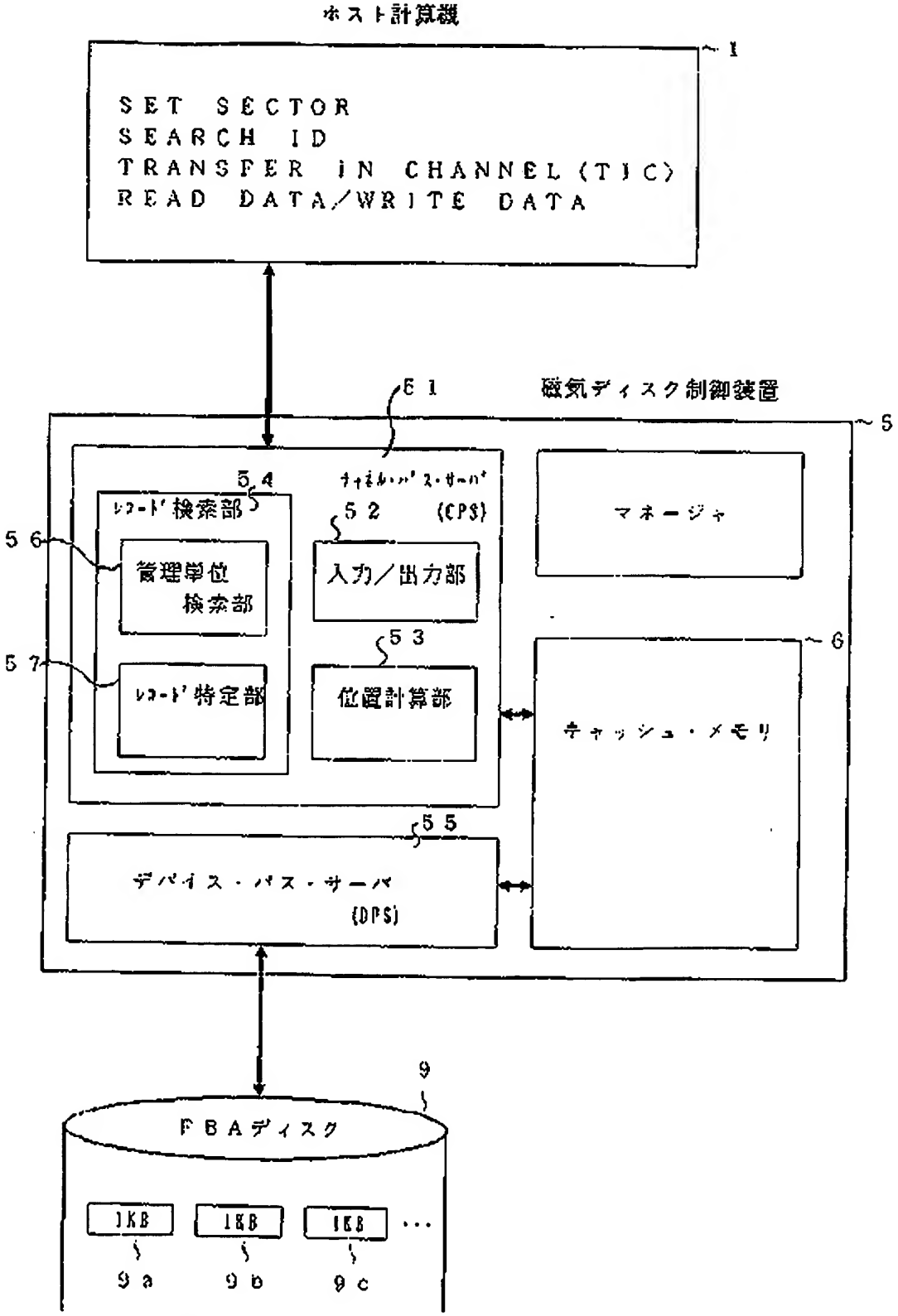
【符号の説明】

１　ホスト計算機
２ａ　チャネル
２ｂ　チャネル
２ｃ　チャネル
２ｄ　チャネル
２ｅ　チャネル
３　ＣＫＤフォーマットのレコード
４　ＣＫＤコマンド
５　磁気ディスク制御装置
６　キャッシュ・メモリ
７　ＣＫＤのトラックイメージ
８　ＦＢＡフォーマットのレコード
９　ＦＢＡディスク
１０　ＦＢＡコマンド
５１　チャネルパスサーバ（ＣＰＳ）
５２　入力部／出力部
５３　位置計算部
５４　レコード検索部
５５　読み書き部
５６　管理単位検索部
５７　レコード特定部
１００　ＣＫＤ－ＦＢＡ変換装置
１０１　ギャップ削除部

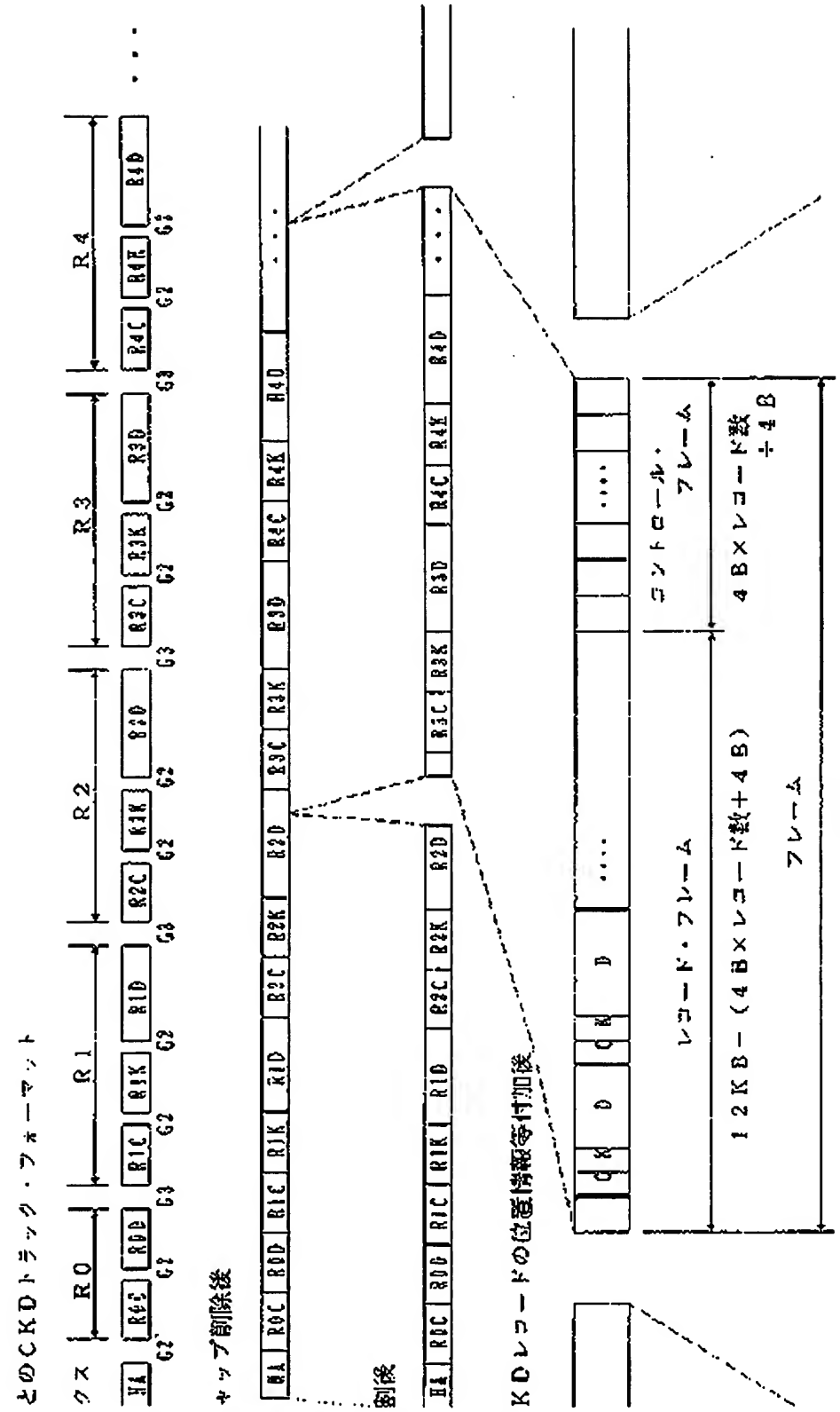
【図1】


1
ホスト計算機
2a
2b
2c
2d
2e
チャネル
チャネル
チャネル
チャネル
チャネル
2
CKDフォーマット
のレコード 3
4 CKDコマンド
3a
3b
3c
磁気ディスク制御装置 5
キャッシュ・メモリ
6
CKDのトラックイメージ
7
CKD－FBA変換
10 FBAコマンド
8a
8b
8c
8d
8e
FBAフォーマット
のレコード 8
（ブロック）
FBAディスク 9
9a
9b
9c
9d
9e
データの流れ

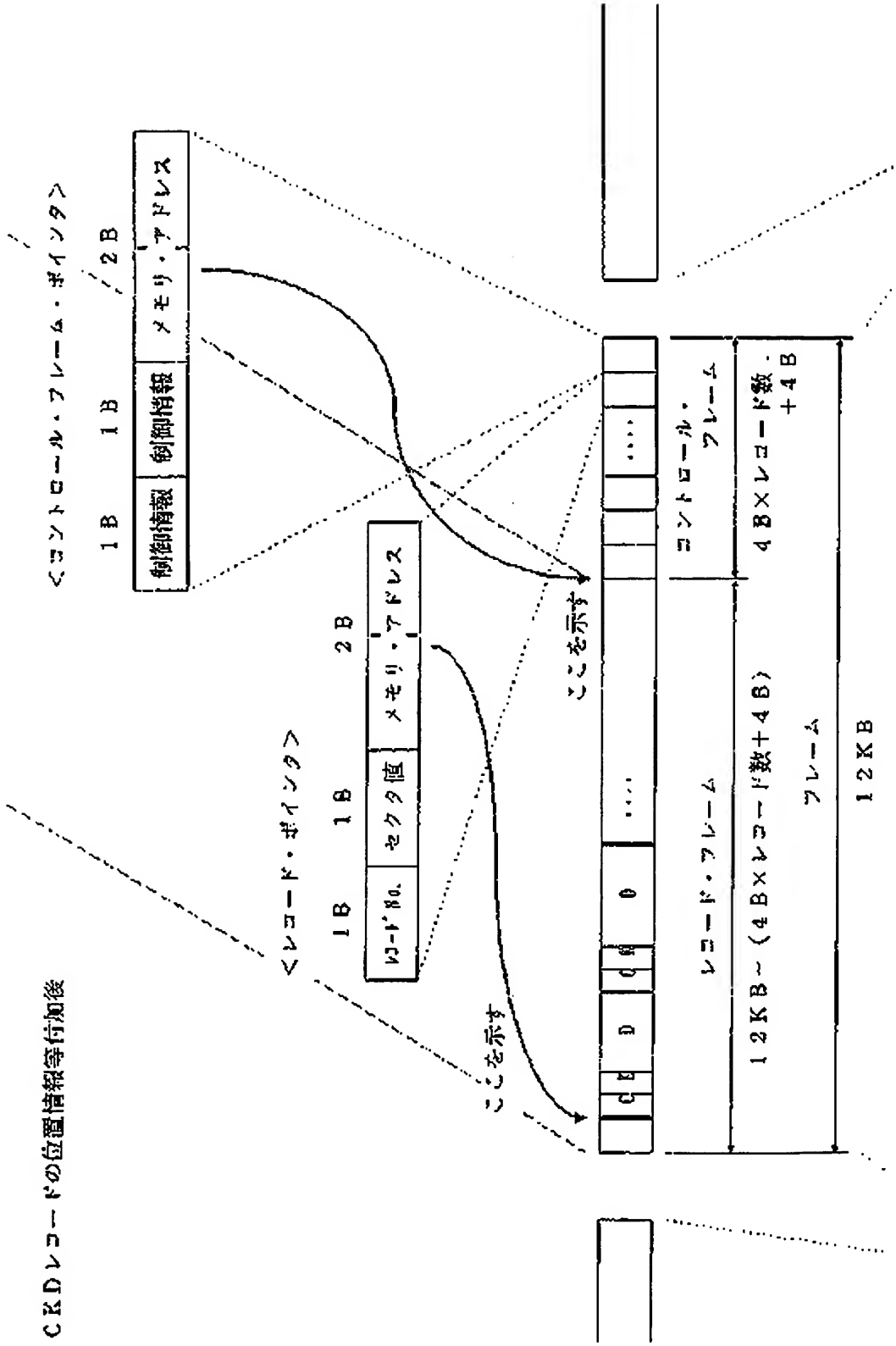
【図2】


ホスト計算機
1
SET SECTOR
SEARCH ID
TRANSFER IN CHANNEL (TIC)
READ DATA/WRITE DATA
51
磁気ディスク制御装置
5
54
レコード検索部
52
(CPS)
マネージャ
56
管理単位
検索部
入力／出力部
53
57
レコード特定部
位置計算部
6
キャッシュ・メモリ
55
デバイス・パス・サーバ
(DPS)
9
FBAディスク
1KB
1KB
1KB
9a
9b
9c

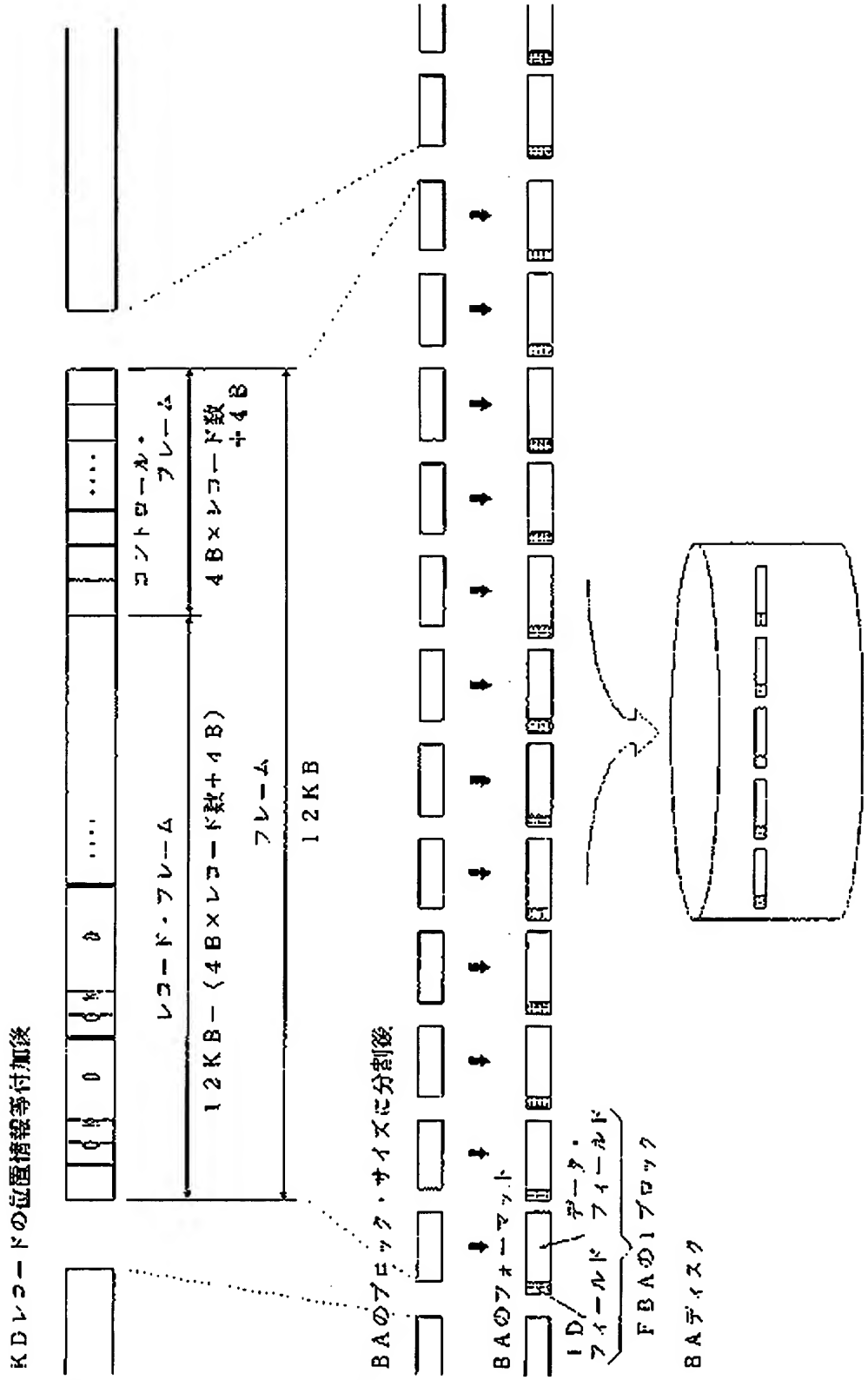
【図3】


とのCKDトラック・フォーマット
クス
R0
R1
R2
R3
R4
HA
R0C
R0D
R1C
R1K
R1D
R2C
R2K
R2D
R3C
R3K
R3D
R4C
R4K
R4D
G1
G2
G3
キャップ削除後
削後
KDレコードの位置情報等付加後
レコード・フレーム
コントロール・フレーム
4B×レコード数÷4B
12KB－（4B×レコード数＋4B）
フレーム
12KB

【図4】


CKDレコードの位置情報等付加後
<レコード・ポインタ>
1B　1B　2B
レコード No.
セクタ値
メモリ・アドレス
ここを示す
<コントロール・フレーム・ポインタ>
1B　1B　2B
制御情報
制御情報
メモリ・アドレス
ここを示す
D
0
コントロール・フレーム
4B×レコード数・+4B
レコード・フレーム
12KB～（4B×レコード数+4B）
フレーム
12KB

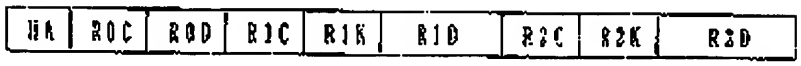
【図5】


KDレコードの位置情報等付加後
レコード・フレーム
12KB－（4B×レコード数＋4B）
フレーム
コントロール・フレーム
4B×レコード数＋4B
12KB
BAのブロック・サイズに分割後
BAのフォーマット
ID・フィールド
データ・フィールド
FBAの1ブロック
BAディスク

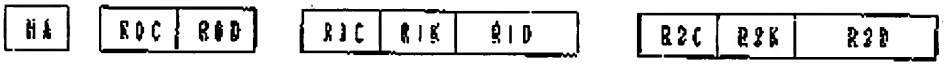
【図6】

＜元のＣＫＤトラック・フォーマット＞

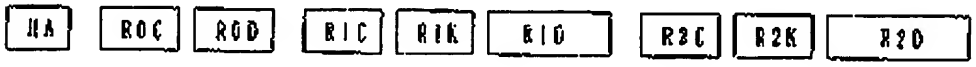
＜ケース　Ｉ－１＞



＜ケース　Ｉ－２＞



＜ケース　Ｉ－３＞



ギャップの残し方

【図16】

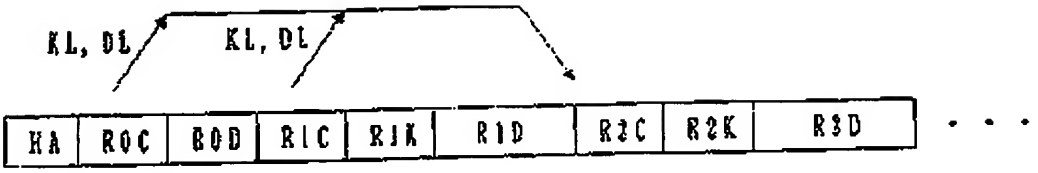
●フォーマット４（最少レコード数のケース）
Ｒ０　：　ＫＬ　＝　０　，　ＤＬ　＝　４７９８８
Ｒｎ　：　無し

この時のＲ０のＤＬの値は、以下のようにして求められる。
図１３に示すように、Ｒ０のデータ・フィールドとして使用できるバイト数は、
４８５１２　－　Ｇ２Ｃ　－　Ｒ０Ｃ　－　Ｇ２　＝
４８５１２　－　２４８　－　４０　－　２３４　＝４８０００

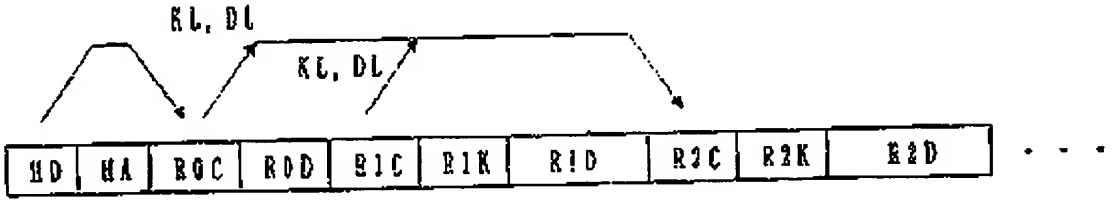
この４８０００ｂｙｔｅのうち　１２ｂｙｔｅはＥＣＣとスペース

【図7】

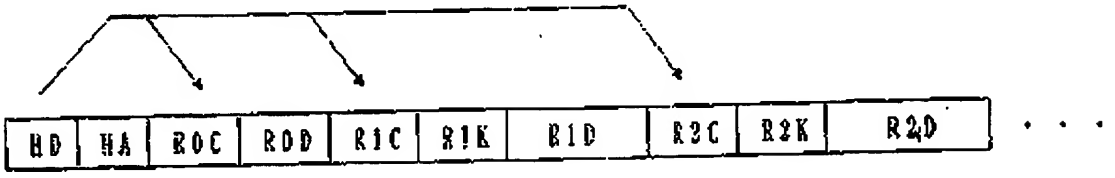
＜ケース　II－1＞



＜ケース　II－2＞

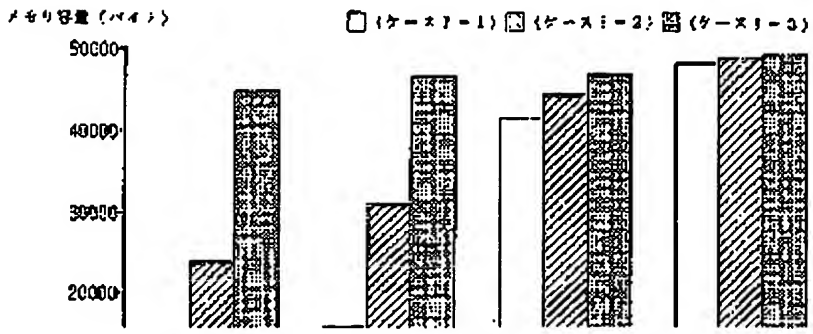


＜ケース　II－3＞

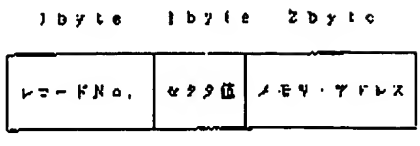


レコードの位置決め方法

【図18】



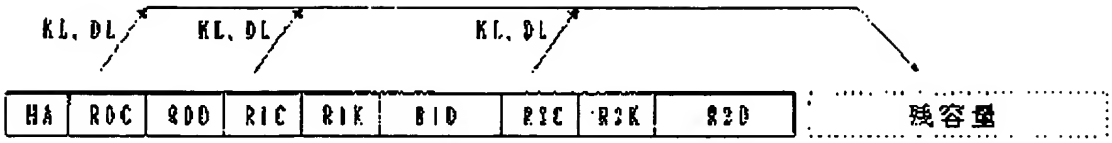
【図20】



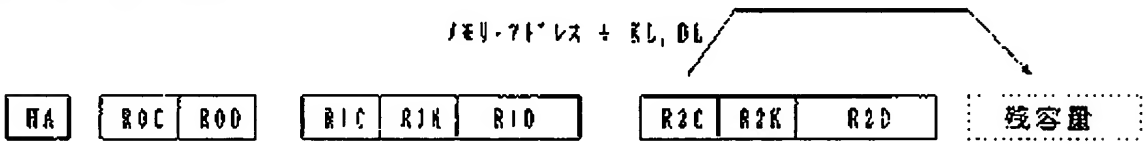
レコード1個当りの登録情報

【図8】

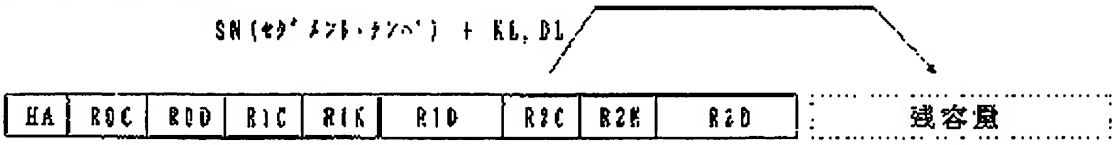
＜ケース　Ⅲ－1＞



＜ケース　Ⅲ－2＞



＜ケース　Ⅲ－3＞



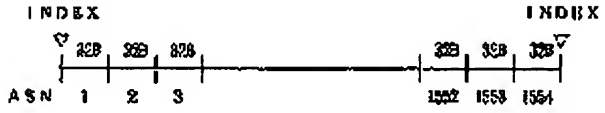
トラック容量のチェック方法

【図21】

仮想情報の管理単位のケース

	仮想情報の管理単位	サイズ
管理単位1	FBAのブロック（セクタ）	2KB
管理単位2	データ・ディスクへの分割単位（1／4トラック）	12KB

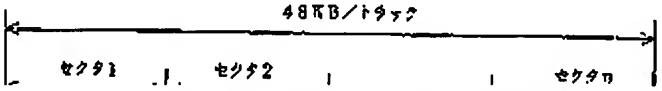
【図29】



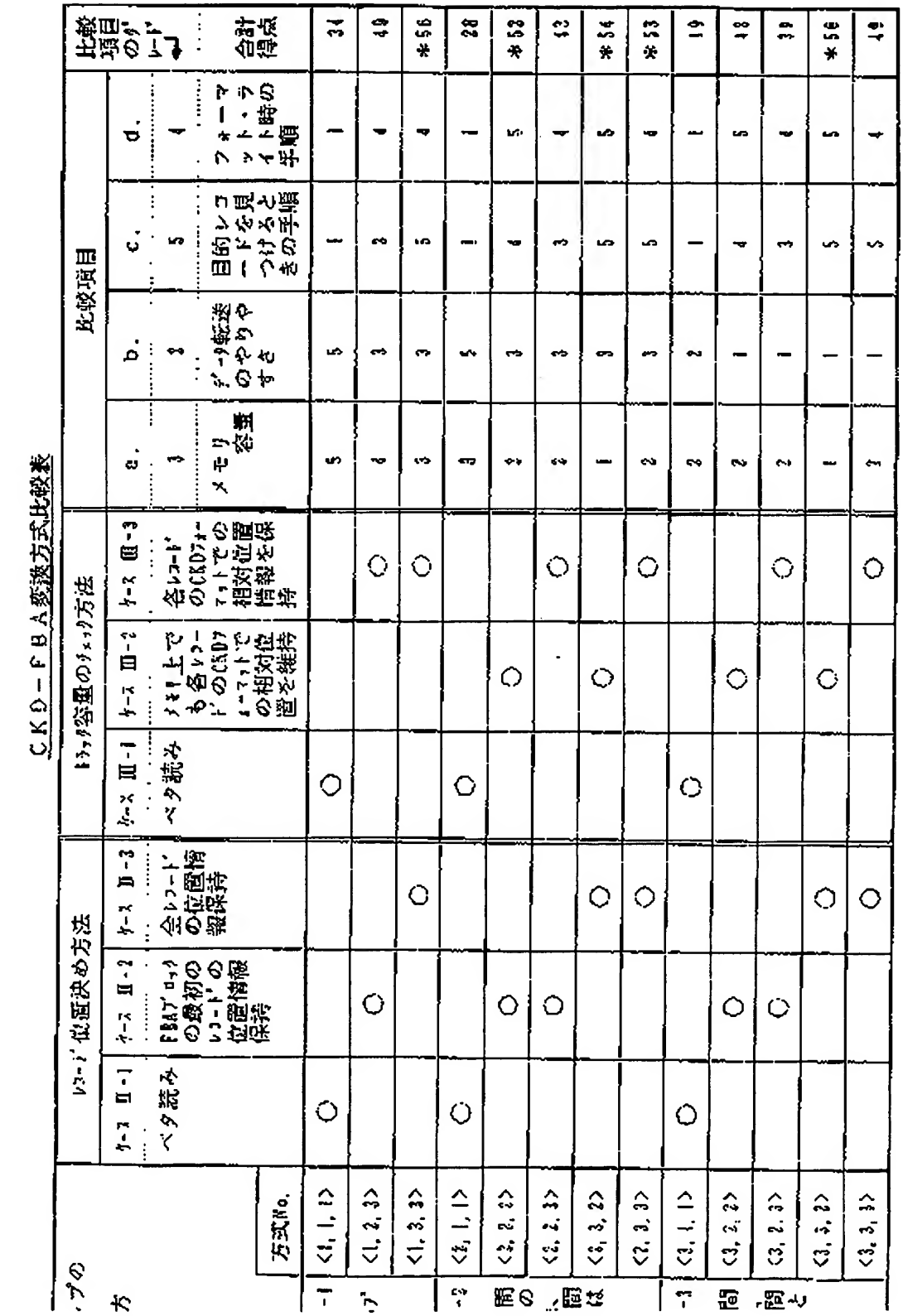
（注）ASN　絶対セグメント番号

セグメント・ナンバ

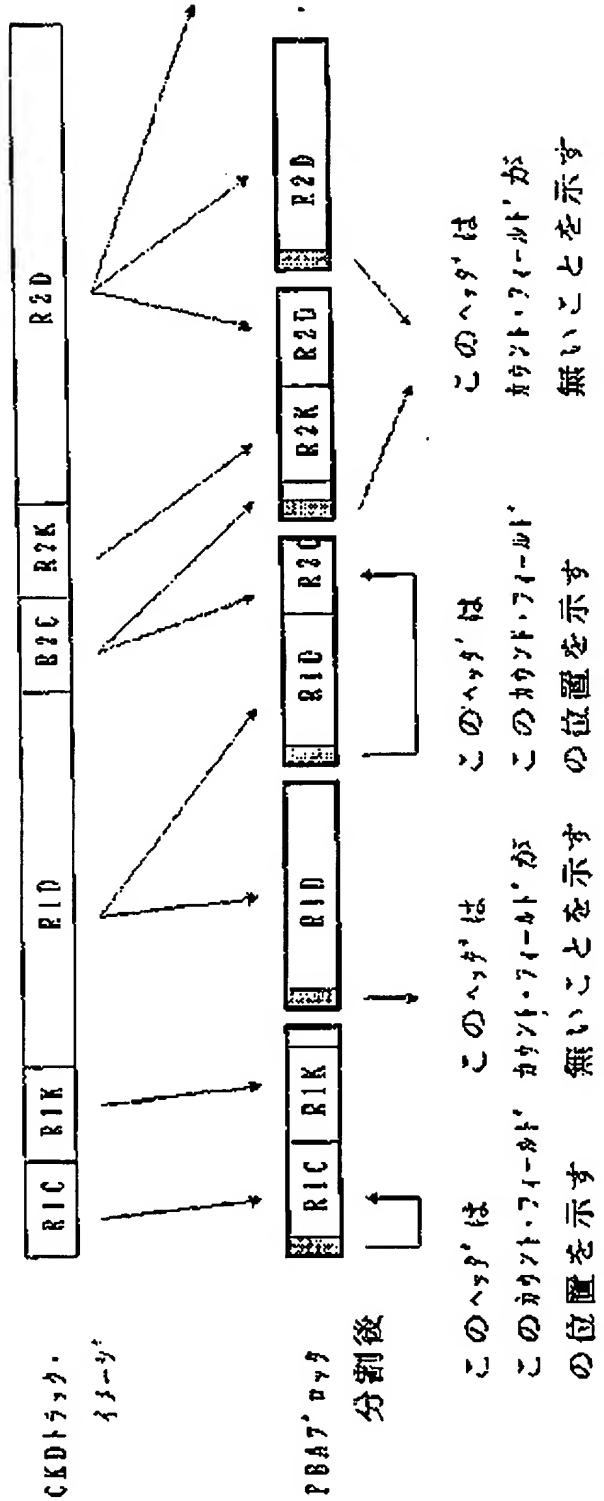
【図35】



【図9】


CKD－FBA変換方式比較表
ケース I-1 ベタ読み
ケース II-2 FBAブロックの最初のレコードの位置情報保持
ケース II-3 全レコードの位置情報保持
ケース III-1 ベタ読み
ケース III-2 メモリ上でも各レコードのCKDフォーマットでの相対位置を維持
ケース III-3 各レコードのCKDフォーマットでの相対位置情報を保持
a. メモリ容量
b. データ転送のやりやすさ
c. 目的レコードを見つけるときの手順
d. フォーマット・ライト時の手順
合計得点
方式No.
<1, 1, 1>
<1, 2, 3>
<1, 3, 3>
<2, 1, 1>
<2, 2, 2>
<2, 2, 3>
<2, 3, 2>
<2, 3, 3>
<3, 1, 1>
<3, 2, 2>
<3, 2, 3>
<3, 3, 2>
<3, 3, 3>
34
40
*56
28
*52
42
*54
*53
19
48
39
*56
40

【図10】



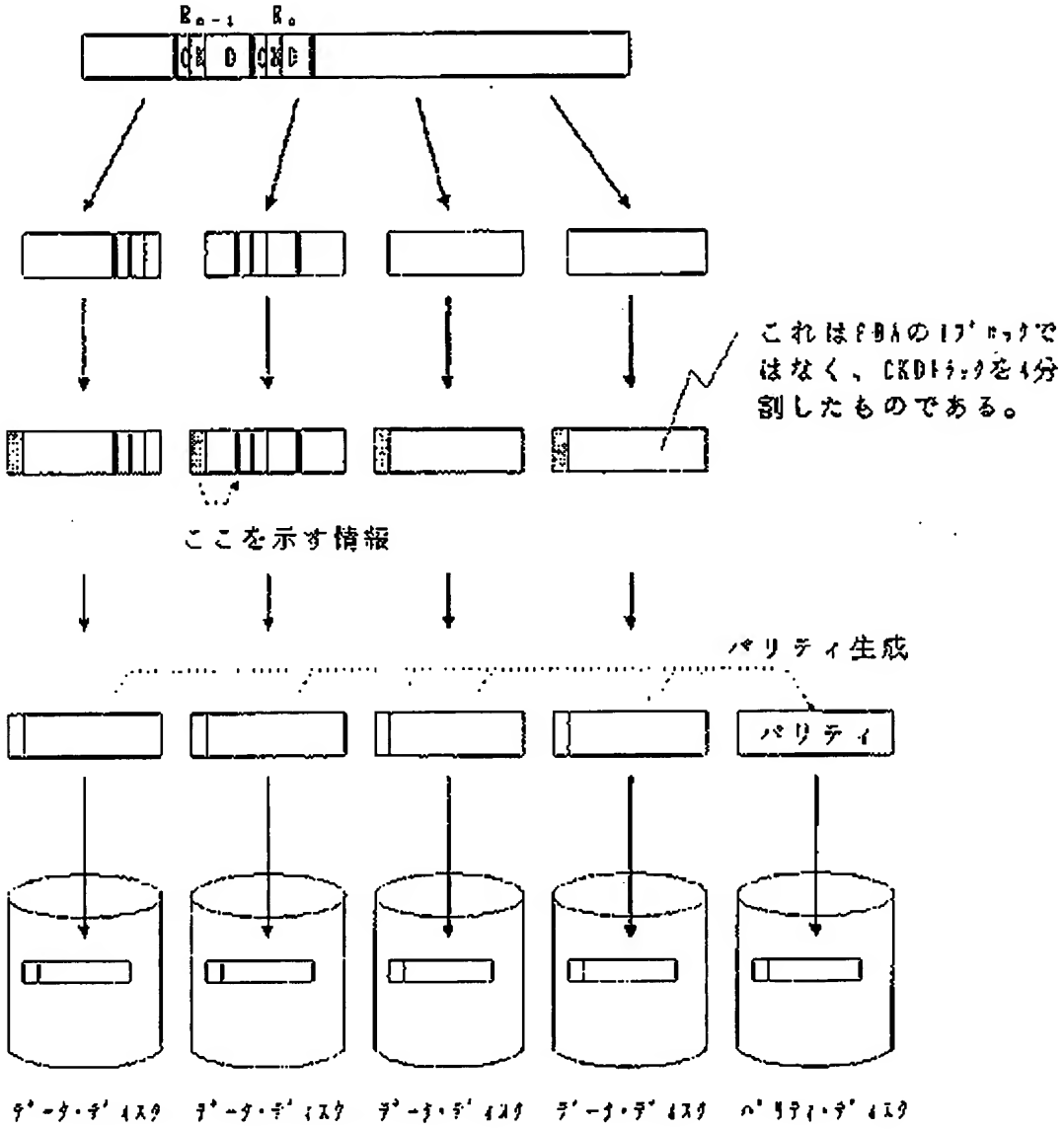
【図31】

CKDレコードのFBAブロック分割例

フィールドごとの最大削除バイト数

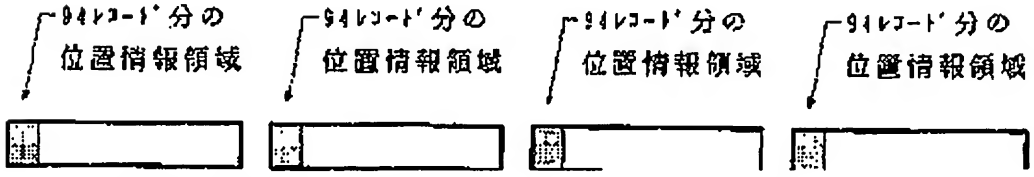
G1	H1	G2C	R0C	G2	R0D	G3	R0C	G2	R0K	G2	R0D
504	12	248	26	224	24	216	26	224	31	224	31
合計 1038										合計 752	

【図１１】



RAIDにおけるＣＫＤトラックの分割

【図２４】



【図12】

メモリ容量見積りケース

見積り ケース	レコード・サイズ (RS) KL	レコード・サイズ (RS) DL	レコード・サイズ (Rs) KL	レコード・サイズ (Rs) DL	レコード数 ノミナル RO	レコード数 ノミナル Rn	備考
フォーマット1	0 B	1B	0 B	1B	1 + 9B		最多レコード数のケース
フォーマット2	0 B	8B	0 B	200B	1 + 6B		シーケンシャル・ファイル の典型
フォーマット3	0 B	8B	0 B	4096B	1 + 1B		ページング、スタッピング、ESAM
フォーマット4	0 B	479888	—	—	1 + 0		最少レコード数のケース

位置情報のメモリ容量見積り

【図27】

	管理単位のサイズ	レコードの位置決め方式	管理単位内のレコード数	管理単位当りの位置情報	トラック当りの位置情報
管理単位2	12KB	<ケース II-2>	Max. 94コ	4B	18B
		<ケース II-3>	Max. 94コ	376B	1504B
		<ケース II-3>改	Max. 94コ	380B	360B

【図13】

●フォーマットI（最多レコード数のケース）

R0 : KL = 0 , DL = 1

Rn : KL = 0 , DL = 1

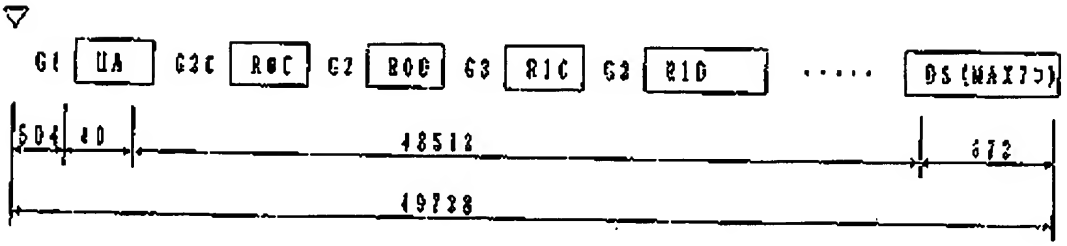
R0のサイズ = G2C + R0C + G2 + R0D
= 248 + 40 + 224 + 32
= 544 (byte)

Rnのサイズ = G3 + RnC + G2 + RnD
= 216 + 40 + 224 + 32
= 512 (byte)

図に示すように、HA以降のトラック容量は 48512byte
であるから、Rnのレコード数は下式のようになる。（割算は切捨て）

$$\text{Rnのレコード数} = \frac{48512 - \text{R0のサイズ}}{\text{Rnのサイズ}}$$

$$= \frac{48512 - 544}{512} = 93$$



単位：バイト

トラック・フォーマット

【図14】

●フォーマット2（ジャーナル・ファイルの典型）

R0 : KL = 0 , DL = 8

Rn : KL = 0 , DL = 200

R0のサイズ = G2C + R0C + G2 + R0D

= 248 + 40 + 224 + 32

= 544（byte）

Rnのサイズ = G3 + RnC + G2 + RnD

= 216 + 40 + 224 + 224

= 704（byte）

図13に示すように、HA以降のトラック容量は 48512byte であるから、Rnのレコード数は下式のようになる。（割算は切捨て）

$$\text{Rnのレコード数} = \frac{48512 - \text{R0のサイズ}}{\text{Rnのサイズ}}$$

$$= \frac{48512 - 544}{704} = 68$$

【図23】

レコードのために必要な最大メモリ容量

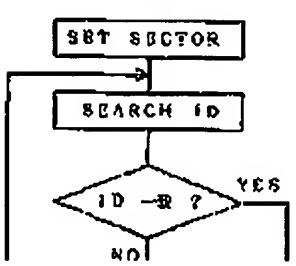
ギャップの削除方法	フォーマット4 / 最少レコード
<ケース I-1> 全ギャップ削除	43044B
<ケース I-2> フィールド間ギャップのみ削除	48796B

【図36】

(a)

SET SECTOR
SEARCH ID
TRANSFER IN CHANNEL (TIC)
READ DATA / WRITE DATA

(b)



【図15】

●フォーマット3（ページング、スワッピング、ＶＳＡＭ）
Ｒ０ ： ＫＬ ＝ ０ ， ＤＬ ＝ ８
Ｒｎ ： ＫＬ ＝ ０ ， ＤＬ ＝ ４０９６

Ｒ０のサイズ ＝ Ｇ２Ｃ ＋ Ｒ０Ｃ ＋ Ｇ２ ＋ Ｒ０Ｄ
＝ ２４８ ＋ ４０ ＋ ２２４ ＋ ３２
＝ ５４４（ｂｙｔｅ）

Ｒｎのサイズ ＝ Ｇ３ ＋ ＲｎＣ ＋ Ｇ２ ＋ ＲｎＤ
＝ ２１６ ＋ ４０ ＋ ２２４ ＋ ４１２８
＝ ４６０８（ｂｙｔｅ）

図１３に示すように、ＨＡ以降のトラック容量は ４８５１２ｂｙｔｅ であるから、Ｒｎのレコード数は下式のようになる。（剰算は切捨て）

$$\text{Rnのレコード数} = \frac{48512 - \text{R0のサイズ}}{\text{Rnのサイズ}}$$

$$= \frac{48512 - 544}{4608} = 10$$

【図28】

レコードのために必要な最大メモリ容量

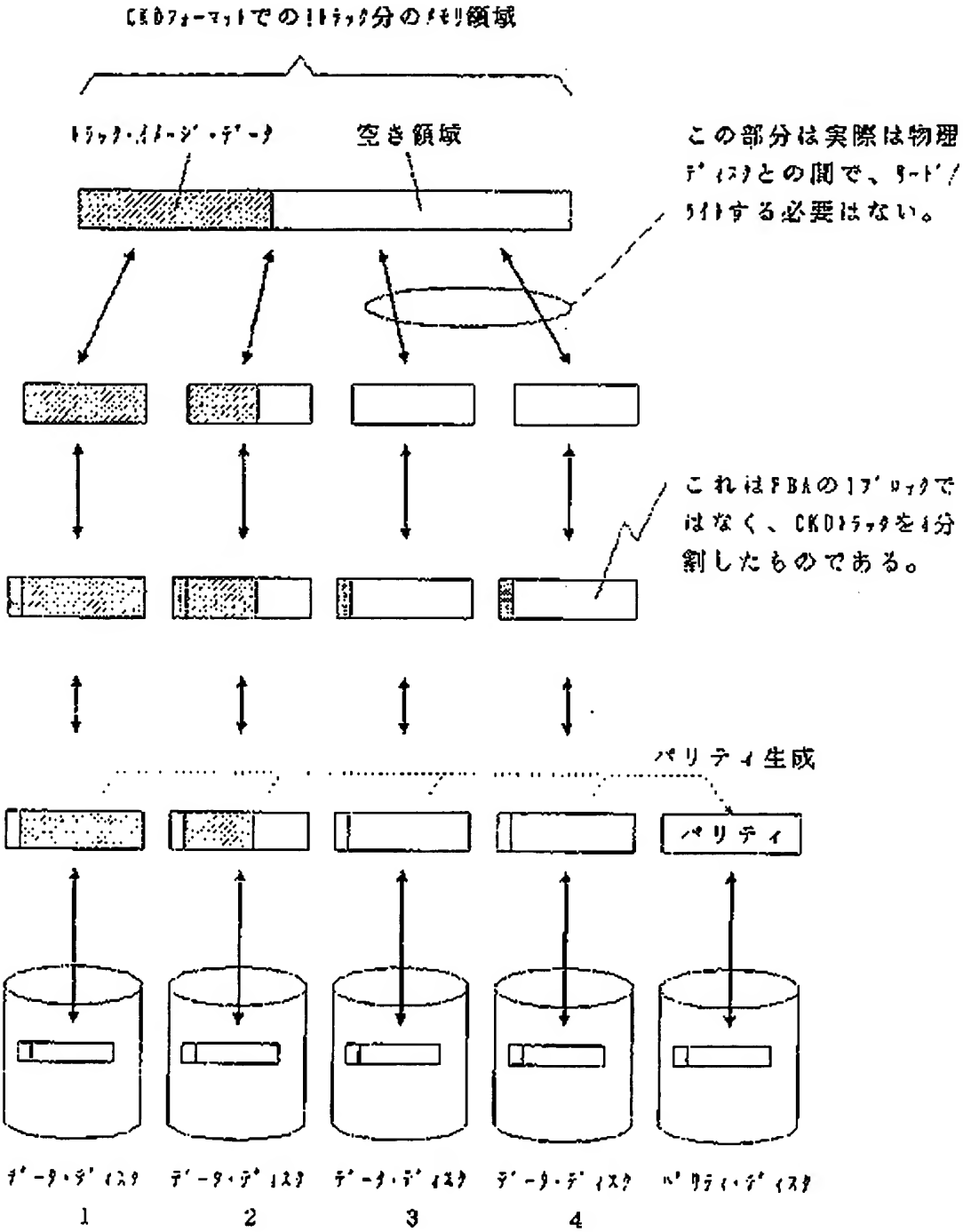
ギャップの削除方法	ﾌｫｰﾏｯﾄ4
	最少ﾚｺｰﾄﾞ
ｹｰｽ Ⅰ-1 全ｷﾞｬｯﾌﾟ削除	480446

【図17】

メモリ容量見積り

	フォーマット1 … 最多レコード	フォーマット2 … RnD = 2006	フォーマット3 … RnD=40965	フォーマット4 … 最少レコード
<ケース I -1> 全ギャップ削除	3754	15568	41304	48044
<ケース I -2> フィールド間ギャップのみ削除	23594	31008	44216	48796
<ケース I -3> 全ギャップ残す	44650	46464	46680	49020

【図19】

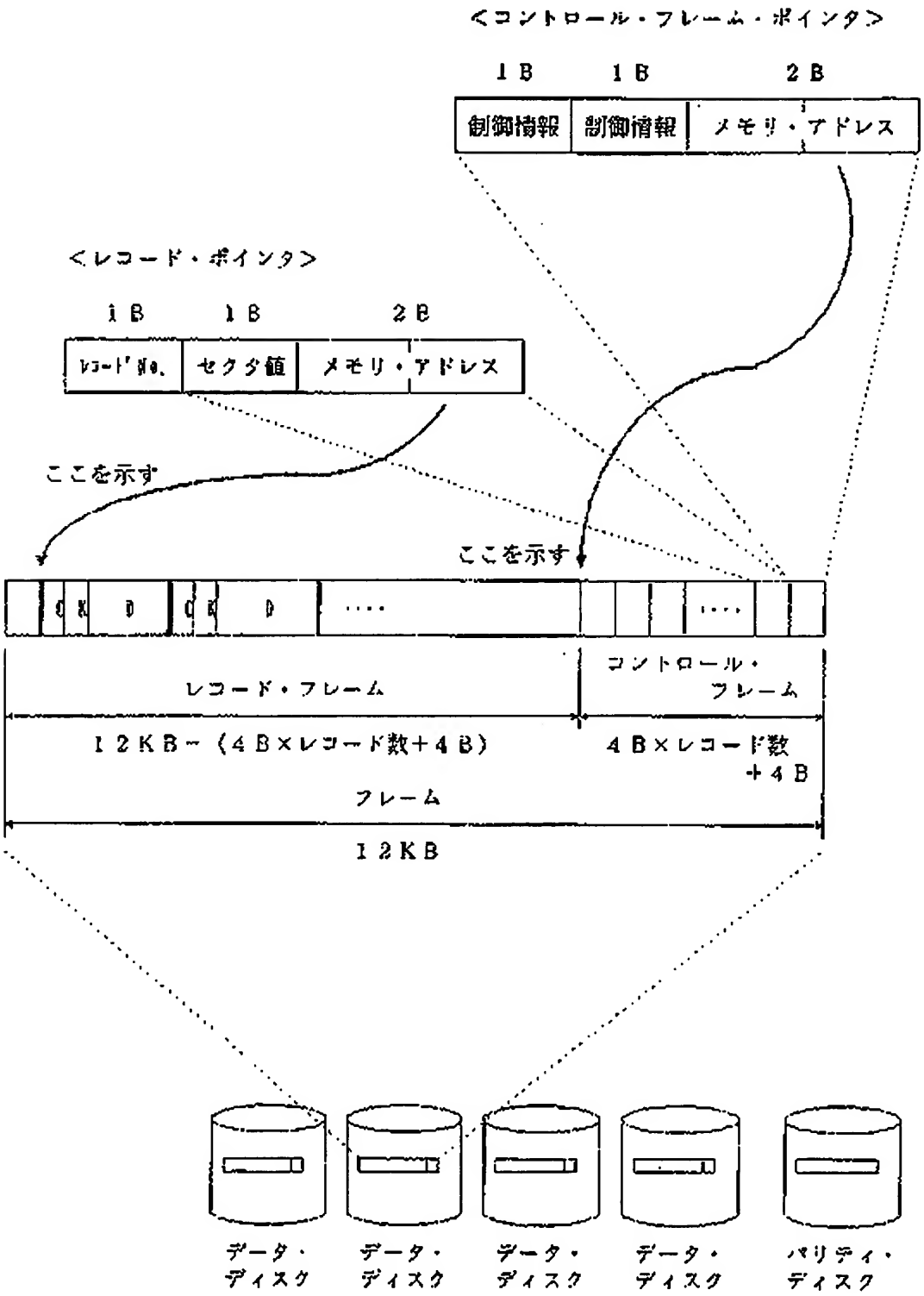

この部分は実際は物理
ディスクとの間で、リード/
ライトする必要はない。
これはFBAのブロックで
はなく、CKDトラックを4分
割したものである。
パリティ生成
パリティ
1
2
3
4

【図22】

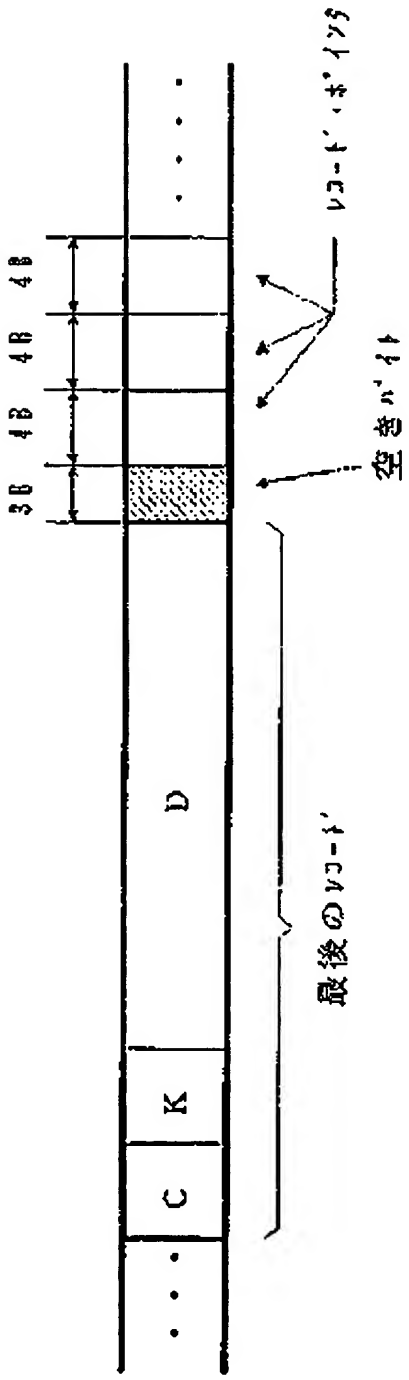
位置情報のメモリ容量見積り

	管理単位のサイズ	レコード'の位置決め方式	管理単位内のレコード'数	管理単位当りの位置情報	トラック当りの位置情報
管理単位1	2KB	<ケース II-2>	Max. 100	4B	96B
		<ケース II-3>	Max. 622	24BB	5952B
管理単位2	12KB	<ケース II-2>	Max. 943	4B	16B
		<ケース II-3>	Max. 943	3168	15048

【図25】

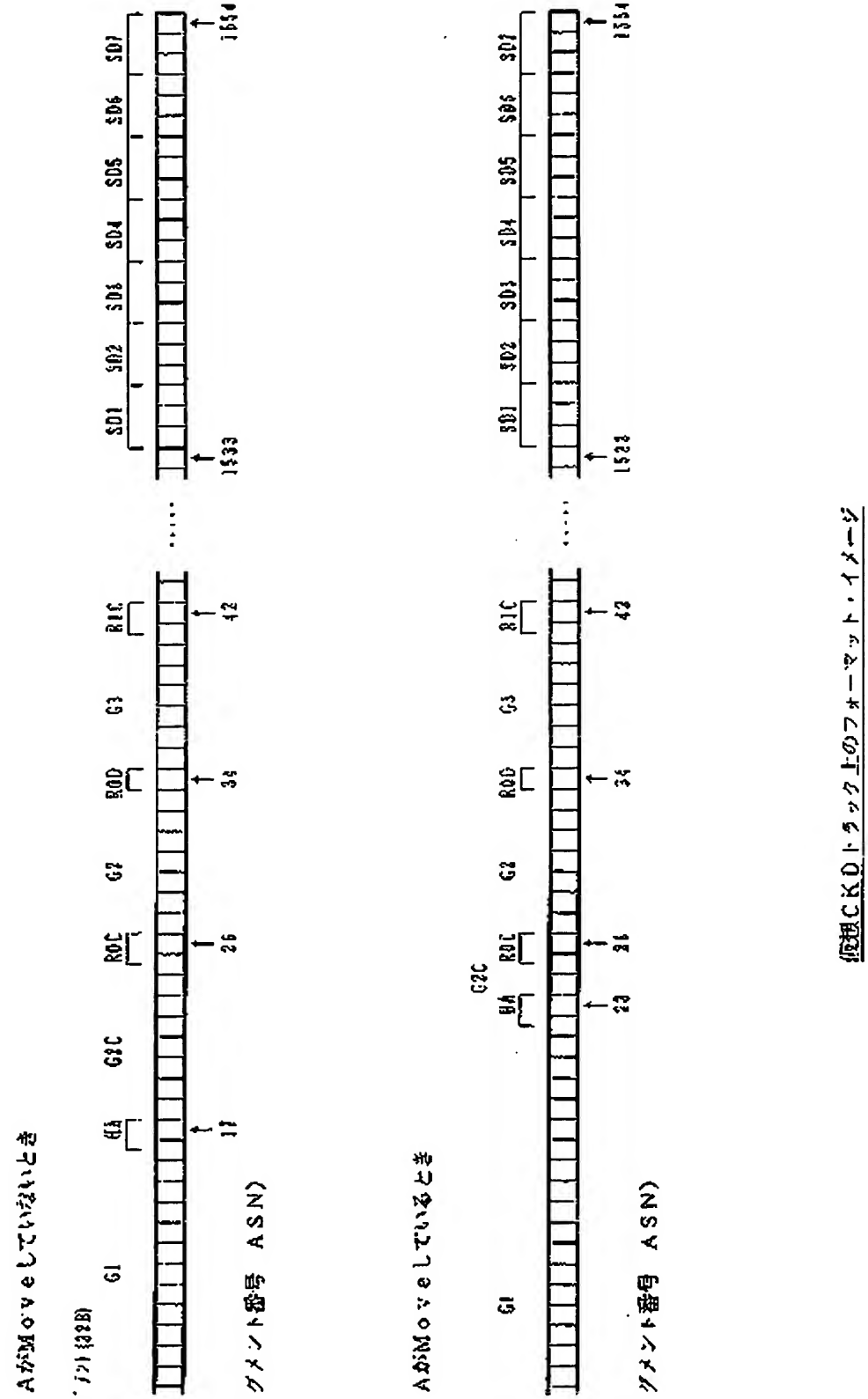

<コントロール・フレーム・ポインタ>
1B
1B
2B
制御情報
制御情報
メモリ・アドレス
<レコード・ポインタ>
1B
1B
2B
レコードNo.
セクタ値
メモリ・アドレス
ここを示す
ここを示す
C
K
D
C
K
D
....
....
レコード・フレーム
コントロール・フレーム
12KB－（4B×レコード数＋4B）
4B×レコード数＋4B
フレーム
12KB
データ・ディスク
データ・ディスク
データ・ディスク
データ・ディスク
パリティ・ディスク

【図26】



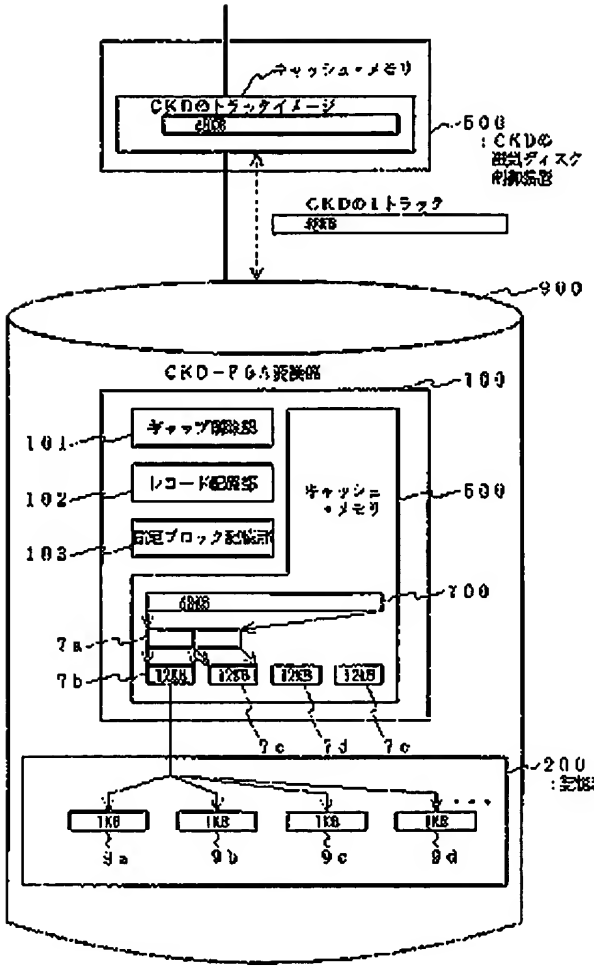
フレーム中の空きバイト

【図30】

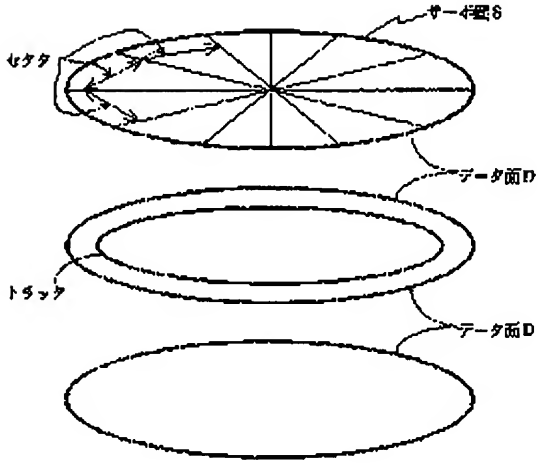


仮想CKDトラック上のフォーマット・イメージ

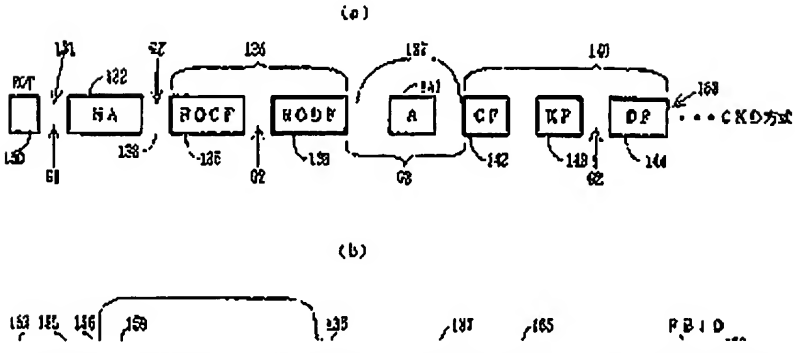
【図３２】


キャッシュ・メモリ
CKDのトラックイメージ
500
：CKDの
磁気ディスク
制御装置
CKDの1トラック
900
CKD－FBA変換部
100
101
102
103
キャッシュ
・メモリ
500
700
7a
7b
7c
7d
7e
200
：記憶部
9a
9b
9c
9d

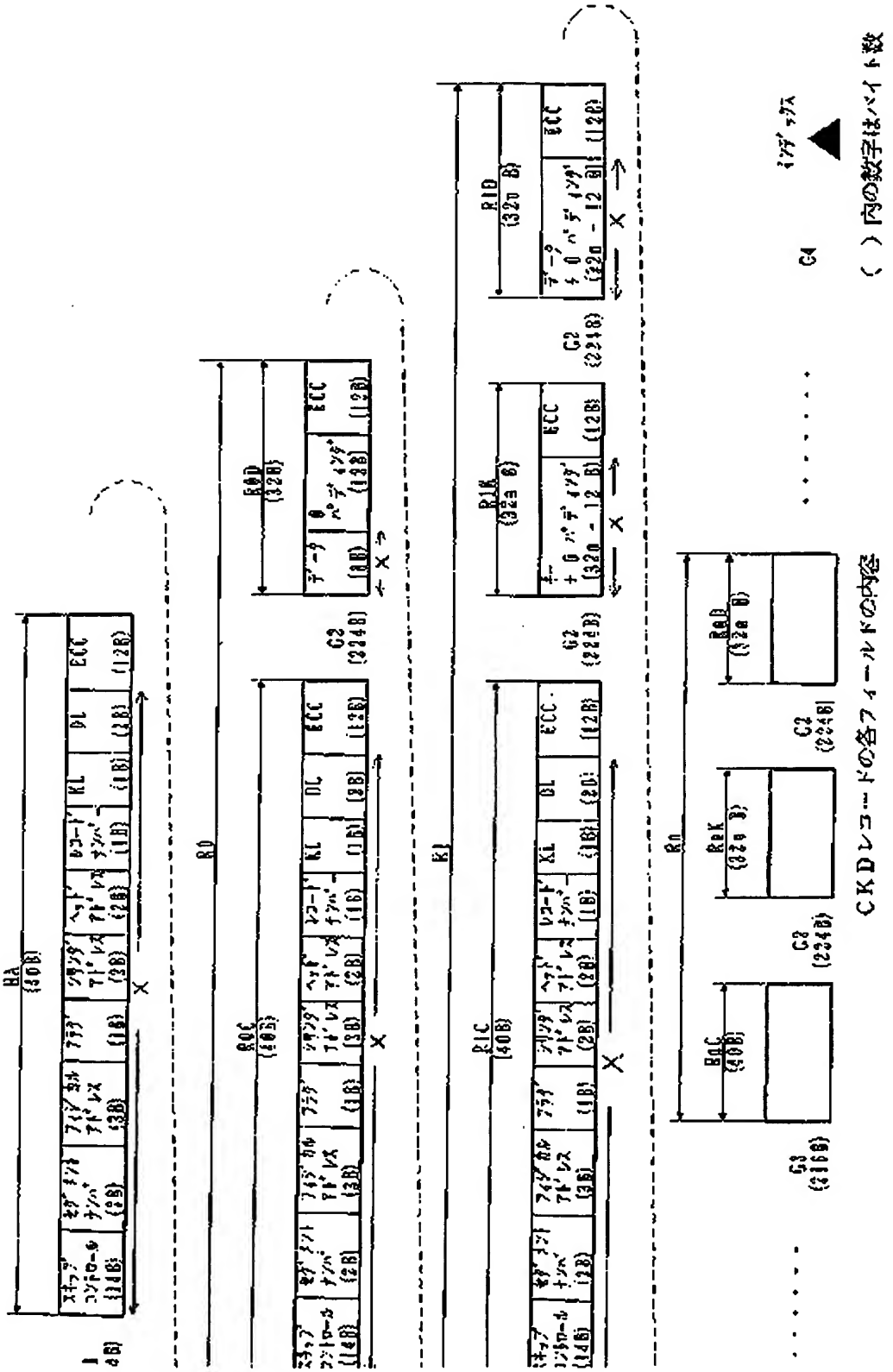
【図３４】


セクタ
データ面D
トラック
データ面D

【図３７】


(a)
RA
ROCF
ROOF
A
CF
KP
DF
・・・CKD方式
(b)

【図33】



CKDレコードの各フィールドの内容

【公報種別】特許法第１７条の２の規定による補正の掲載
【部門区分】第６部門第３区分
【発行日】平成１０年（１９９８）１１月４日

【公開番号】特開平５－３０７４４０
【公開日】平成５年（１９９３）１１月１９日
【年通号数】公開特許公報５－３０７５
【出願番号】特願平５－９００２
【国際特許分類第６版】
G06F 3/06 301
G11B 20/12
【ＦＩ】
G06F 3/06 301 V
301 Y
G11B 20/12

【手続補正書】
【提出日】平成９年４月１８日
【手続補正１】
【補正対象書類名】明細書
【補正対象項目名】０００２
【補正方法】変更
【補正内容】
【０００２】
【従来の技術】従来、汎用計算機システムの外部記憶装置としては、可変長レコード（ＣＫＤ；Ｃｏｕｎｔ　Ｋｅｙ　Ｄａｔａ）方式の固定ディスク装置が広く用いられている。可変長レコード方式とは、磁気ディスク上の１トラックに記録する物理的なレコードの長さも数も可変である記録フォーマットである。一般的に可変長レコード方式の磁気ディスクの１トラックは図３３に示すようにインデックス・マークから始まり、まず最初にホーム・アドレス（ＨＡ）と引き続くレコード・ゼロ（Ｒ０）が存在する。このＨＡとＲ０はそのトラックのシリンダ・アドレスやヘッド・アドレスなどのアドレス情報以外に、そのトラック上の磁気的な傷の位置や、不良トラックであるか否かの情報、また、不良トラックであれば、交代割り付けされている相手先のトラックのアドレス等の制御情報を含んでいるため、必ず存在しなくてはならない。このＨＡ、Ｒ０に引き続くレコード１（Ｒ１）以降のレコードが通常のデータを記録するレコードで、レコード１個１個の長さも個数もその総容量がトラで、シリンダ・アドレス（ＣＣ）、ヘッド・アドレス（ＨＨ）、レコード・ナンバ（Ｒ）などのアドレス情報のほかに、そのレコードのキー・フィールドの長さ（ＫＬ）、データ・フィールドの長さ（ＤＬ）などを含んでいるので、引き続くキー・フィールド、データ・フィールドの長さを自由に変えることができる。キー・フィールド、データ・フィールドがそのレコードの実際のデータを記録する部分であるが、そのレコードのキーを記録すべきキー・フィールドが不要な場合は、カウント・フィールドでＫＬ＝０を指定することにより、キー・フィールドの存在しないレコードを作ることもできる。ＨＡ、Ｒ０、Ｒ１、Ｒ２・・・などの間には、適当なサイズのレコード間ギャップ（Ｇ１，Ｇ２，Ｇ３）が設けられており、特にＲ１，Ｒ２・・・などのレコードの前のギャップ（Ｇ３）には、アドレス・マーク（ＡＭ）と呼ばれる磁気的に特別なマークが記録されているため、トラック内のどの部分から読み始めても、このアドレス・マークを検出することにより、レコードの読み始めを見つけることができる。
【手続補正２】
【補正対象書類名】明細書
【補正対象項目名】００３３
【補正方法】変更
【補正内容】
【００３３】次に、図２に戻り磁気ディスク制御装置５

２を有している。入力部５２からホスト計算機１で動作しているソフトウェアによる可変長レコード方式のアクセス命令が入力された場合に位置計算部５３は例えばセットセクタコマンドで指定されたセクタ値に基づいてアクセスする可変長レコードが記憶されていると考えられる管理単位の位置を計算する。この位置計算部による計算は前述したようにＦＢＡディスク９にデータが記録される場合には全てのギャップが取り除かれているためにＣＫＤディスク１０に存在している場合よりも前方向に詰まった形でデータが記憶されており．位置計算部５３は経験により．アクセスしようとしてる可変長レコードが記憶されていると考えられる管理単位の位置を計算する。レコード検索部５４は位置計算部５３により計算された管理単位の位置をもとに管理単位をキャッシュメモリ６に読み込みアクセスしようとするレコードを検索する。管理単位検索部５６はアクセスしようとするレコードの相対位置情報（セクタ値）と検索された管理単位に保持されている相対位置情報（セクタ値）とを比較し、アクセスしようとする相対位置情報をもつ管理単位が見つかるまでＦＢＡディスク９に存在する管理単位を順にキャッシュメモリに読み込む。レコード特定部５７は管理単位検索部５６により目的とする管理単位が検索された場合にアクセスしようとする可変長レコードをその管理単位に含まれるアドレス情報を用いて特定する。ＦＢＡディスク９とキャッシュ・メモリ６の間のデータの読み書きをＦＢＡコマンドを使用して実行するのは．データ・パス・サーバ（ＤＰＳ）５５である。マネージャは．全体の制御とキャッシュ・メモリ６の管理を行なう。

【手続補正３】
【補正対象書類名】明細書
【補正対象項目名】００６１
【補正方法】変更
【補正内容】
【００６１】メモリ容量を見積もるに当り、ここでは、下記のような条件で見積りを行う。
（１）トラック・フォーマットとしては図３３を使用する。
（２）各フィールドのＥＣＣ，３２Ｂ境界にあわせるための０パディングは削除する。
（３）ＨＡ、カウント・フィールドのその他のパラメータはすべて残す。
（４）レコードの位置情報などのＣＫＤ－ＦＢＡ変換の

【手続補正４】
【補正対象書類名】明細書
【補正対象項目名】００６３
【補正方法】変更
【補正内容】
【００６３】＜ケース　Ｉ－２＞
・レコード間ギャップだけ残し、フィールド間ギャップは取り除く（ＥＣＣやパディングなどは取り除く）。ただし．フィールド間ギャップやＥＣＣ，パディングなどを削除してもその分をギャップ長を伸ばしてつじつまを合わせることにすると、メモリ容量は元のＣＫＤフォーマットと全く同じになる。従って、ここでは削除分のつじつま合わせは行わないケースで見積りを行う。
●フォーマット１（最多レコード数のケース）
ＨＡのサイズ　＝Ｇ１＋ＨＡ＝５０４＋（４０－１２）＝５３２
Ｒ０のサイズ　＝Ｇ２Ｃ＋Ｒ０Ｃ＋Ｒ０Ｄ＝２４８＋（４０－１２）＋１＝２７７
Ｒｎのサイズ　＝Ｇ３＋ＲｎＣ＋ＲｎＤ＝２１６＋（４０－１２）＋１＝２４５
よって、１トラック分の容量は、
ＨＡ＋Ｒ０＋Ｒｎ×９３＝５３２＋２７７＋２４５×９３＝２３５９４
●フォーマット２（ジャーナル・ファイルの典型）
ＨＡのサイズ　＝Ｇ１＋ＨＡ＝５０４＋（４０－１２）＝５３２
Ｒ０のサイズ　＝Ｇ２Ｃ＋Ｒ０Ｃ＋Ｒ０Ｄ＝２４８＋（４０－１２）＋８＝２８４
Ｒｎのサイズ　＝Ｇ３＋ＲｎＣ＋ＲｎＤ＝２１６＋（４０－１２）＋２００＝４４４
よって、１トラック分の容量は、
ＨＡ＋Ｒ０＋Ｒｎ×６８＝５３２＋２８４＋４４４×６８＝３１００８
●フォーマット３（ページング、スワッピング．ＶＳＡＭ）
ＨＡのサイズ　＝Ｇ１＋ＨＡ＝５０４＋（４０－１２）＝５３２
Ｒ０のサイズ　＝Ｇ２Ｃ＋Ｒ０Ｃ＋Ｒ０Ｄ＝２４８＋（４０－１２）＋８＝２８４
Ｒｎのサイズ　＝Ｇ３＋ＲｎＣ＋ＲｎＤ＝２１６＋（４０－１２）＋４０９６＝４３４０
よって、１トラック分の容量は、
ＨＡ＋Ｒ０＋Ｒｎ×１０＝５３２＋２８４＋４３４０×

ＨＡ＋Ｒ０＝５３２＋４８２６４＝４８７９６
【手続補正５】
【補正対象書類名】明細書
【補正対象項目名】００６４
【補正方法】変更
【補正内容】
【００６４】＜ケース　Ｉ－３＞
・レコード間ギャップだけでなく、すべてのフィールド間ギャップも残す（ＥＣＣやパディングなどは取り除く）。この場合も、ＥＣＣ，パディングなどを削除してもその分をギャップ長を伸ばしてつじつまを合わせることにすると、メモリ容量は元のＣＫＤフォーマットと全く同じになる。従って、ここでは削除分のつじつま合わせは行わないケースで見積りを行う。
●フォーマット１（最多レコード数のケース）
ＨＡのサイズ　＝Ｇ１＋ＨＡ=５０４＋（４０－１２）＝５３２
Ｒ０のサイズ　＝Ｇ２Ｃ＋Ｒ０Ｃ＋Ｇ２＋Ｒ０Ｄ＝２４８＋（４０－１２）＋２２４＋１＝５０１
Ｒｎのサイズ　＝Ｇ３＋ＲｎＣ＋Ｇ２＋ＲｎＤ＝２１６＋（４０－１２）＋２２４＋１＝４６９
よって、１トラック分の容量は、
ＨＡ＋Ｒ０＋Ｒｎ×９３＝５３２＋５０１＋４６９×９３＝４４６５０
●フォーマット２（ジャーナル・ファイルの典型）
ＨＡのサイズ　＝Ｇ１＋ＨＡ=５０４＋（４０－１２）＝５３２
Ｒ０のサイズ　＝Ｇ２Ｃ＋Ｒ０Ｃ＋Ｇ２＋Ｒ０Ｄ＝２４８＋（４０－１２）＋２２４＋８＝５０８
Ｒｎのサイズ　＝Ｇ３＋ＲｎＣ＋Ｇ２＋ＲｎＤ＝２１６＋（４０－１２）＋２２４＋２００＝６６８
よって、１トラック分の容量は、
ＨＡ＋Ｒ０＋Ｒｎ×６８＝５３２＋５０８＋６６８×６８＝４６４６４
●フォーマット３（ページング、スワッピング，ＶＳＡＭ）
ＨＡのサイズ　＝Ｇ１＋ＨＡ=５０４＋（４０－１２）＝５３２
Ｒ０のサイズ　＝Ｇ２Ｃ＋Ｒ０Ｃ＋Ｇ２＋Ｒ０Ｄ＝２４８＋（４０－１２）＋２２４＋８＝５０８
Ｒｎのサイズ　＝Ｇ３＋ＲｎＣ＋Ｇ２＋ＲｎＤ＝２１６＋（４０－１２）＋２２４＋４０９６＝４５６４
よって、１トラック分の容量は、
ＨＡ＋Ｒ０＋Ｒｎ×６８＝５３２＋５０８＋４５６４×６８＝４８４８８＝４９０２０
【手続補正６】
【補正対象書類名】明細書
【補正対象項目名】００７３
【補正方法】変更
【補正内容】
【００７３】次に、「レコードの位置情報の制御方式の改善」について説明する。先の見積りの制御方式の問題点は以下のとおりである。前述した「レコードの位置情報のメモリ容量見積り」では、ＣＫＤレコードの位置情報を管理する単位(レコードの位置情報を保持する単位)ごとに、そこに記録され得る最大レコード数に合わせて位置情報の領域を確保していた。具体的にいうと，図２４に示すように，管理単位のサイズを１２ＫＢとしたときは，そこに最大９４個のＣＫＤレコードが記録され得る。レコード１個当りの位置情報としては４ｂｙｔｅを見積もっていたので、１２ＫＢの管理単位ごとに
４Ｂ×９４＝３７６Ｂ
だけの位置情報領域を見積もっていた。ＣＫＤトラック１本分のトラック・イメージはこの１２ＫＢの管理単位４個からなっており，ＣＫＤトラック１本当りの位置情報のメモリ容量は、
３７６Ｂ×４＝１５０４Ｂ
となってしまう。しかし，これでは，ＣＫＤトラック１本当り９４×４＝３７６レコード分の位置情報の領域を確保していることになる。図３３のフォーマットでは、１ＣＫＤトラックの最大レコード数は９４レコード（Ｒ０含む）であるので、１２ＫＢの各管理単位に記録され得る最大レコード数がそれぞれ９４個であっても，同時に全部の管理単位に９４個ずつのレコードが記録されることはあり得ない。従って，９４×３＝２８２レコード分の位置情報の領域は無駄になっていることになる。
【手続補正７】
【補正対象書類名】明細書
【補正対象項目名】００８０
【補正方法】変更
【補正内容】
【００８０】次に、この改善案による位置情報のメモリ容量の見積りを行う。この改善案による位置情報のメモリ容量について，前述した「レコードの位置情報のメモリ容量見積り」で見積もったケースのうち図２２の管理単位２のケースについて見積りを行い，改善効果を見

とメモリ容量の評価」参照）。このとき、本改善案では、一つ目のフレームのコントロール・フレームにレコード・ポインタが９４個、コントロール・フレーム・ポインタが１個あって、他のフレームに制御情報は特にない（というより、他のフレームそのものが要らない）。従って、位置情報の制御に必要なメモリ容量は、

４Ｂ×９５＝３８０Ｂ

である。この結果と、「レコードの位置情報のメモリ容量見積り」で見積もった他のケースとの比較を図２７に示す。＜ケース　ｉｉ－２＞は従来の方式で、各ＦＢＡブロック（ここでは、フレーム）の先頭に位置するレコードの位置だけを保持する方式である。＜ケース　ｉｉ－３＞はすべてのレコードの位置情報を保持する方式で、本改善前の方式である。＜ケース　ｉｉ－３＞改が本方式である。

【手続補正８】
【補正対象書類名】明細書
【補正対象項目名】００９２
【補正方法】変更
【補正内容】
【００９２】１．ＳＥＴ　ＳＥＣＴＯＲ
＜機能＞チャネルから１バイトのセクタ情報を受け取り、セクタを検索する。（セクタ情報：Ｘ'００'～Ｘ'ＤＤ'，Ｘ'ＦＦ'ただしＸ'　'は１６進数を表す。）
＜実現方式＞このディスク・サブシステムでは、セクタ＊値は、まず、ＣＫＤトラックを四つに分割したフレームのうち、どのフレームに目的レコードが存在するかを知るために使用する。フレームが見つかった後は、コントロール・フレーム中のレコード・ポインタに書かれている各レコードのセグメント・ナンバと、このセクタ値を比較することにより、より精度の高い位置決めができる（フレーム、及びコントロール・フレームやレコード・ポインタの概略については図２５に示したが、詳細は「レコードの位置情報の制御方式の改善」参照のこと）。
○目的レコードが存在するフレームの計算
実際のＣＫＤディスクでは、セクタ１個分は２２４バイトからなりたっているので、セクタ値からＣＫＤフォーマットのトラック上でのそれに相当する絶対バイト・アドレスはただちに計算できる。しかし、フレーム上ではギャップやＥＣＣ、パディングなどは削除して記録しているので、その絶対バイト・アドレスがどのフレームに含まれているかは、その削除してしまった分を換算しないと分からない。この削除した分量はレコード数にも依存する。

【手続補正９】
【補正対象書類名】図面
【補正対象項目名】図１６
【補正方法】変更
【補正内容】
【図１６】

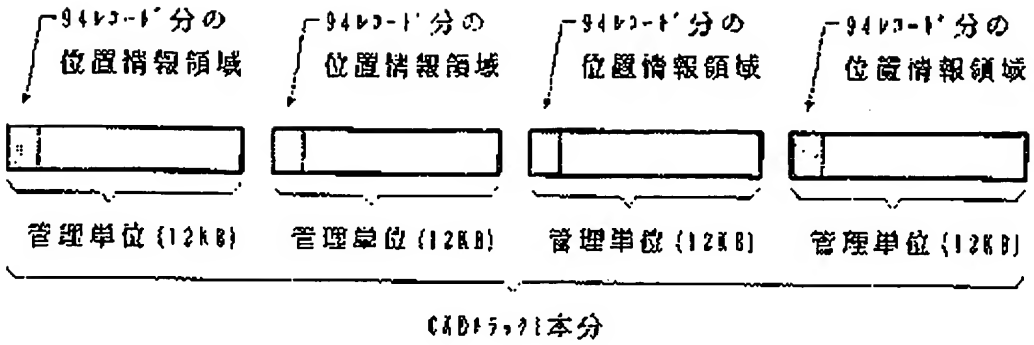
●フォーマット４（最少レコード数のケース）
Ｒ０：ＫＬ＝０，ＤＬ＝４７９８８
Ｒｎ：無し

この時のＲ０のＤＬの値は、以下のようにして求められる。
図１３に示すように、Ｒ０のデータ・フィールドとして使用できるバイト数は、
４８５１２－Ｇ２Ｃ－Ｒ０Ｃ－Ｇ２＝
４８５１２－２４８－４０－２２４＝４８０００

この４８０００ｂｙｔｅのうち、１２ｂｙｔｅはＥＣＣでとられるので、結局、Ｒ０Ｄのデータとして使用できる最大値は、

４８０００　１２＝４７９８８

である。



「レコードの位置情報のメモリ容量見積り」での位置情報の保持方法